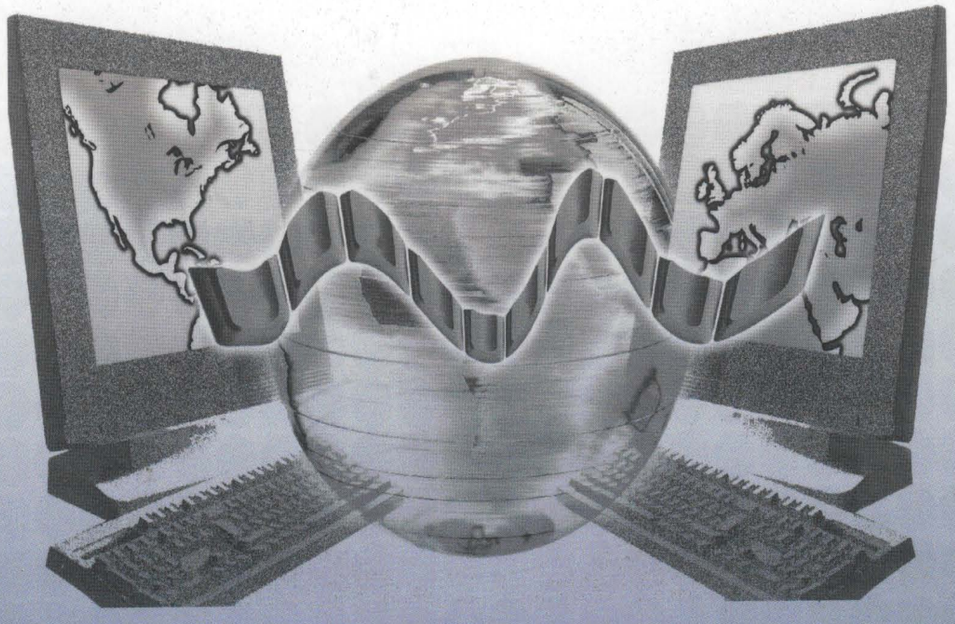
2006 International Symposium
on Distributed Computing and Applications
to Business, Engineering and Science

DCABES 2006 PROCEEDINGS

Volume I

Editor in Chief: Xu Wenbo
Associate Editor: Wang Guangming



SHANGHAI UNIVERSITY PRESS

2006 International Symposium
on Distributed Computing and Applications
to Business, Engineering and Science

DCABES 2006 PROCEEDINGS

Volume I

Editor in Chief: Xu Wenbo
Associate Editor: Wang Guangming

SHANGHAI UNIVERSITY PRESS

图书在版编目（CIP）数据

2006年国际电子、工程及科学领域的分布式计算应用学术研讨会论文集 / 须文波主编. —上海：上海大学出版社，2006.10
ISBN 7-81118-023-5

Ⅰ.2… Ⅱ.须… Ⅲ.电子计算机—国际学术会议—文集—英文 Ⅳ.TP3-53

中国版本图书馆CIP数据核字(2006)第120341号

责任编辑 姚铁军
封面设计 孙 敏

DCABES2006 PROCEEDINGS

2006年国际电子、工程及科学领域的分布式计算和应用学术研讨会论文集

须文波 主编

王光明 副主编

会议承办单位：江南大学
会议协办单位：浙江工商大学



上海大学出版社出版发行
（上海市上大路99号 邮政编码 200444）
（http://www.shangdapress.com 发行热线 66135110）

出版人：姚铁军

*

宜兴市德胜印刷有限公司印刷
开本 1/16 印张 89.5 字数 1 878千
2006年10月第一版 2006年10月第一次印刷
印数：001-600册
ISBN 7-81118-023-5/TP.037

定价：300.00元

DCABES 2006 PROCEEDINGS

Volume I

责任编辑 姚铁军

封面设计 孙 敏

ISBN 7-81118-023-5

ISBN 7-81118-023-5/TP · 037

Price: 300 RMB

CONTENTS

Volume I

1. Parallel and Distributed Algorithms

2. Distributed Computing and Distributed Systems

3. Clustering Computing, Grid Computing and parallel processing

4. Image, Vision and Voice Processing

5. Intelligent computing

6. Mobile Computing

7. Distributed Database and Data Mining

Preface

The series of meetings, International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), is now becoming an important international event on various applications and the related computing environments of distributed and grid computing. The first meeting was held at Wuhan University of Technology, Wuhan, and the second meeting was held at Southern Yangtze University, Wuxi, the third meeting was held at Wuhan University of Technology, Wuhan, the fourth meeting was held at Greenwich University, Greenwich. In this year, the fifth meeting will be organized by Southern Yangtze University and Zhejiang GongShang University and hold at Hangzhou. The conference theme include not only its traditional theme such as parallel and distributed computing, but also intelligent and high performance computing that will be described as follows.

It was my pleasure that the DCABES2006 conference had received a great number of papers submitted cover a wide range of topics, such as Parallel/Distributed Algorithms, Distributed System and Distributed Computing, Grid Computing and Parallel Processing, Network and Applications, Database and Engineering Applications.

Papers submitting to the conference come from over 16 countries and regions. All papers contained in this Proceeding are peer-reviewed and carefully chosen by members of Scientific Committee, proceeding editorial board and external reviewers. Papers accepted or rejected are based on majority opinions of the referee's. All papers contained in this proceeding give us a glimpse of what future technology and applications are being researched in the distributed computing area in the world.

I would like to thank all members of the Scientific Committee, the local organizer committee, the proceedings editorial board and external reviewers for selecting the papers. Special thanks are due to Dr. Choi-Hong LAI, Prof. Qingping Guo and Prof. GuangMing Wang, who co-chaired the Scientific Committee with me. It is indeed a pleasure to work with them and obtain their suggestions. I am also grateful to Professor H. Power, Professor Peter M.A. Sloot, Professor Aoying Zhou, Professor Yao Zhen, for their contributions of keynote speeches in the conference.

Sincerely thanks should be forwarded to the China Ministry of Science and Technology (MOST), the China Ministry of Education (MOE), the Natural Science Foundation of China (NSFC) and Southern Yangtze University and Zhejiang GongShang University.

Finally I should also thank Professor Yun Ling, WeiMing Wang and Mrs Li Liu, for their efforts in conference organizing activities, my postgraduate students, such as Mr. Jun Xu, Mr. Peng Wang, and Mr. ZhaoKuo Nan, for their time and help. Without their time and efforts this conference cannot be organized smoothly.

Enjoy your stay in Hangzhou. Hope to meet you again at the DCABES 2007.

Professor Wenbo Xu,
Chair of the DCABES2006
School of Information Technology
Southern Yangtze University
Jiangsu, China

Chair of Scientific Committee

Xu, Prof. W. B., Southern Yangtze University, Wuxi, China

Co-Chair of Scientific Committee

Guo, Prof. Q. P., Wuhan University of Technology, Wuhan, China
Lai, Prof. C. -H., University of Greenwich, UK
Wang, Professor G. -M., Zhejiang Gongshang University, China

Chair of Organizeing Committtee

Xu, Prof. W. B., Southern Yangtze University, Wuxi, China

Scientific Committee (in alphabetical order)

Cai,Prof. X. C., University of Colorado, Boulder, U.S.A

Cai,Professor Jiamei, Zhejiang Industry University, Hangzhou, China

Cao,Prof. J.W., Research and Development Centre for Parallel Algorithms and Software, Beijing, China

Chi, Prof. X.B., Academia Sinica, Beijing, China

Guo, Professor Q.P., Wuhan University of Technology, Wuhan, China

Ho, Dr. P. T., University of Hong Kong, Hong Kong, China

Jesshope, Professor C., University of Amsterdam, the Netherlands

Kang, Professor L.S., Wuhan University, China

Keyes, Professor D.E., Columbia University, USA

Lai, Prof. C. H., University of Greenwich, UK

Lee, Dr. John, Hong Kong Polytechnic, Hong Kong, China

Liddell, Professor H. M., Queen Mary, University of London, UK

Lin, Dr. H.X., Delft University of Technology, Delft, the Netherlands

Lin, Dr. P., National University of Singapore, Singapore

Loo, Dr. Alfred, Hong Kong Lingnan University, Hong Kong, China

Sloot, Professor P.M.A., University of Amsterdam, Amsterdam the Netherlands

Sun, Professor J., Academia Sinica, Beijing, China

Tsui, Mr. Thomas, Chinese University of Hong Kong, Hong Kong, China

Wang, Professor G. M., Zhejiang Gongshang University, China

Wang, Professor Shitong, S.T Southern Yangtze University, Wuxi, China

Wang, Professor WeiMing, Zhejiang Gongshang University, Hangzhou, China

Xu, Professor W.B., Southern Yangtze University, Wuxi, China

Zhang, Professor J., University of Kentucky, USA

Zhuang Professor, Yueting, Zhejiang University,Hangzhou, China

Local Organizing Committee (in alphabetical order)

Chen, Professor Jian, Southern Yangtze University, Wuxi, China

Feng, Professor Bin, Southern Yangtze University, Wuxi, China

Ling, Professor Yun, Zhejiang Gongshang University, Hangzhou, China

Liu, Professor Yuan, Southern Yangtze University, Wuxi, China

Sun, Professor Shouqian, Zhejiang University, Hangzhou, China

Editorial Board

Parallel Multipolar algorithm for the Indirect Boundary Element Numerical solution of Stokes Flow Problems

H. Power
School of Mechanical, Materials and Manufacturing Engineering, The University of Nottingham, University Park, Nottingham, NG7 2RD, United Kingdom
Email: henry.power@nottingham.ac.uk

ABSTRACT

A multipolar expansion technique is applied to the indirect Boundary Element Method formulation in order to solve Stokes Flow problems. Due to the nature of the algorithm, it is necessary to resort to the use of an iterative solver for the resulting algebraic linear system of equations. In comparison with the direct BEM formulation, the indirect formulation is more stable with iterative solvers, and does not need to be preconditioned to obtain a fast rate of convergence. A parallel implementation is designed to take advantage of the natural domain decomposition of fast multipolar techniques and bring further improvement. A good result in memory saving and computing time is obtained that enable us to run huge examples which are prohibitive for traditional BEM implementations.

Keywords: Stokes flow, parallel multipolar algorithm

1. INTRODUCTION

The Boundary Element Method (BEM) is a numerical method of solution of boundary integral equations, based on a discretisation procedure. The basis of the method is to transform the original Partial Differential Equation (PDE), or system of PDEs, that define a given physical problem into an equivalent integral equation (or system) by means of the corresponding Green's representation formula (direct method), or in terms of continuous distribution of singular solutions of the PDE over the boundaries of the problem (indirect method). In this way, the obtained integral equation satisfies the governing field equation exactly, and one seeks to satisfy the imposed boundary conditions approximately.
In comparison with domain methods such as the FEM and FDM, in the BEM the number of degrees of freedom for a given problem is relatively small, due to the boundary only nature of the scheme. However, a main drawback inherent in the BEM is that, dense linear systems must be solved. If a direct solver is used to solve the system of equations, $O(N^3)$ operations and $O(N^2)$ memory are required which limits the BEM applications to medium size problems due to the computational costs of storing and solving such dense systems.

Several O (N) and O (N log N) numerical algorithms are known in the literature to compute the potential and force fields resulting from the interaction of N particles (N particles problem), for which standard solution leads to a computational complexity of $O(N^2)$. These algorithms were based on the expansion of the potential field generated by N sources in multipole or Taylor series and grouping far-field influences.

These methods had immediate implications on BEM for potential problems, as its discrete linear system is the product of pairwise interaction between sources. It allowed in this way the fast evaluation of the integral terms that constitute the BEM matrixes. Nevertheless, due to the nature of influence grouping it is not possible to have an explicit linear system and therefore the use of iterative solvers was required.

Fast Multipole algorithm is also well suited for parallelisation due to the regularity of the data structure and the locality of its data dependencies. It is based on well-defined domain decomposition where the possible communications between clusters is known.

2. STOKES FLOW

Consider the problem of determining the low Reynolds number motion of an incompressible viscous fluid. Under this condition, the velocity field and pressure (**u**,p) satisfy, as a first approximation, the Stokes system of equations for all x belonging to Ω, with dynamic viscosity μ:

$$\mu \frac{\partial^2 u_i}{\partial x_j^{\,2}} = \frac{\partial p}{\partial x_i} \quad \text{and} \quad \frac{\partial u_i}{\partial x_i} = 0$$

Boundary conditions are given in terms of surface velocity, traction or combination between them according to the problem considered.

3. DIRECT INTEGRAL FORMULATION

Green's integral representation formulae analogous to those employed in Potential theory exist for Stokes flow and this use can be traced back to the work of Lorentz [1]. This representation formula has been extensively used in the numerical solution of Stokes problems since the original work of Youngren & Acrivos [2] (direct approach). The application of this approach to the case of the first boundary value problem (Dirichlet type) yields a first kind Fredholm integral equation for the unknown surface traction.

As is known, Fredholm integral equations of the first kind generally give rise to unstable numerical schemes based upon discretization of the surface involved, the instability manifesting itself in the ill conditioning of the matrix approximation of the kernel. Nonetheless, it is possible to apply the discretization method if only low order accuracy is desired and the system of linear equations to be solved is not too ill conditioned, as appears to be the case in those works that have used the Youngren and Acrivos method. On the other hand, solving a well-posed second kind Fredholm integral equation is always stable.

For the case of mixed boundary value problems, the application of the direct approach yields a mixed system of integral equations of the first and second kinds. No general mathematical theory is available in the literature for the analysis of such systems of integral equations.

Although there is an increasing interest in the analysis of the performance of iterative solutions of the equation sets arising from the direct BEM formulation no simple

algorithm has yet been proposed to overcome the lack of convergence when dealing with large problems. Most of the successful reported schemes require suitable matrix transformations and preconditioning to obtain an accurate result for such large problems. Even in such successful cases, the required number of iterations required to obtain a desirable accuracy remains very large and this increases as the number of unknowns (degrees of freedom) increases.

4. INDIRECT INTEGRAL FORMULATION

The main idea of the indirect formulation is to define an integral representation formula that produces a well-posed second kind integral equation, i.e. it is uniquely solvable and possesses a bounded inverse operator and its analytical solution is given in terms of a Neumann series (regular or modified), which is a Picard iteration. It is not always possible to achieve this with the direct formulation due to the rigid structure of the Green's formula.

A major difficulty encountered with the indirect formulation is the necessity of defining a different formulation for each type of boundary value problem, in order to obtain a uniquely solvable integral equation of the second kind, which can be used as the basis of a robust numerical scheme. Generally this is not straightforward, and to prove the well posedness of the resulting integral equation then a formal analysis of the corresponding integral operators is required.

For an exterior Stokes' flow problem with a prescribed boundary velocity, i.e. the first kind boundary value problem (Dirichlet type), Power and Miranda [3] observed that the double layer representation, which leads to a second kind integral equation coming from the jump property of its velocity field across the density carrying surface, can represent only those flow fields corresponding to surfaces which are force and torque free. The representation may be completed by adding terms that give arbitrary total force and torque in suitable linear combinations. Karrila and Kim [4] call Power and Miranda's new method the Completed Double Layer Boundary Integral Equation Method, since it involves the idea of completing the deficient range of the double layer potential. Botte and Power [5] extended Power and Miranda's indirect formulation for the external Dirichlet problem to the internal one, by looking at the non-flux condition of an internal incompressible flow instead of the force and torque conditions of an external flow.

Using similar ideas, Power [6] extended the idea of the complete double layer approach to the case of an exterior Neumann Stokes' flow problem (given surface traction), in which the internal domain is allowed to shrink (air bubble in a viscous fluid), by completing the deficient range of the integral operator obtained from the surface traction of a single layer representation. The resulting integral operator was completed by adding a harmonic potential source at the centre of the bubble, which when associated with a constant pressure is a regular exterior Stokes flow field that introduced the possibility of compressibility of the internal gas.

More recently, Briceño and Power [7] developed a novel indirect second kind integral equation formulation for the solution of the mixed boundary value problem that defines the slow deformation of a viscous drop with several internal solid inclusions. At the drop surface the fluid surface traction is given, while it is considered that each of the interior inclusions moves with a prescribed rigid body motion.

In contrast with the direct approach where it is not possible to guaranty the behavior of an iterative solver, an iterative solution of a well-posed Fredholm equation of the second kind to fixed precision is bounded and it is independent of the number of surfaces nodes (for more details seed Greengard et al. [8]).

The operation count of such iterative approaches is O ($M \times N^2$), where M is the number of iterations. If is possible to define an integral equation formulation that guarantee M remains small, even for large degrees of freedom, N, then the implementation of an iterative solver to obtain the solution of such formulations will substantially improve the performance of the BEM solution of the problem. Gomez and Power [9] compared the behaviour of the GMRES (k) iterative solver without preconditioning, when it was applied to solve the algebraic system of equations obtained with the completed second kind integral equation formulation for the interior Dirichlet Stokes' flow problem and when it was used to solve the algebraic system obtained with the classical direct formulation of the same problem. Gomez and Power's analysis shows that the number of iterations for the direct formulation increases with the number of surface points, reaching a prohibitive value for even a moderate number of nodes, namely five hundred iterations for $1.8x10^3$ surface nodes, while the indirect formulation keeps the number of iterations very small and almost constant, always being less than 20 iterations even for more than $3x10^4$ surface nodes (see Fig. 1).

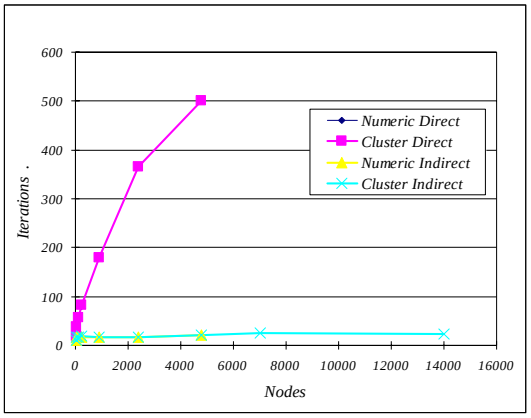


Fig. 1. Number of iterations to solve the linear system of equations for the direct and indirect BEM formulation of exterior Dirichlet problem.

5. MULTIPOLE AND CLUSTERING

Barnes and Hut [10] did much of the early work with truncated multipole expansions. Their tree-code algorithm uses an oct-tree data structure to hierarchically subdivide the simulation domain into well-separated areas, which can interact via the truncated expansions. Their method reduces the computational complexity of the problem from an O (N^2) to O(N log N). On the other hand, Greengard and Rokhlin [11] introduced the concept of local expansion to translate and sum the effects of multiple remote multipole expansions into a single local value. By including these local expansions

with-in an oct-tree data structure, the complexity of the evaluation was further reduced to O (N).

Expanding the kernel in terms of spherical harmonic series and using the duality principle between the inner and outer expansions of harmonic functions Greengard and Rokhlin [12] were able to achieve an O(N) efficiency. In general, this type of efficiency can not be obtained by using Taylor series expansions, since in order to have such a high degree of efficiency it is necessary to perform some kind of local expansion, which should be able to translate the far-fields to single local value, as it is the case in the harmonic analysis due to the dual principle.

Popov and Power [13] presented a multipole BEM strategy developed for 3D elasticity problems which is based on Taylor expansions but requires only *O(N)* operations and *O(N)* memory. Popov and Power's efficient algorithm results from the use of a clustering technique, first shift, in combination with an additional Taylor series expansion around the collocation points, second shift.

The starting point of the multipole scheme presented in this work is the Taylor series expansion of the integral kernel. For example, a single layer integral can be approximated by the following series:

$$\int_{\Gamma} U_{ij}(x,y)\theta_j(y)d\Gamma(y) \sim \sum_{n=1}^{N}\left\{\sum_{\alpha=1}^{m}\theta_j^{n\alpha}U_{ij}(x-y_0^n)\mathrm{M}_0^{n\alpha}\right\}$$
$$+\sum_{n=1}^{N}\left\{\sum_{\alpha=1}^{m}\theta_j^{n\alpha}\sum_{\beta=1}^{\infty}\left[U_{ij,k_1\Lambda k_\beta}(x-y_0^n)\mathrm{M}_{k_1\Lambda k_\beta}^{n\alpha}\right]\right\} \quad (1)$$

where

$$\mathrm{M}_0^{n\alpha} = \int_{\Delta\Gamma_n} N^{\alpha}(\xi,\eta)J(\xi,\eta)d\xi d\eta \quad (2\text{-a})$$

$$\mathrm{M}_{k_1\Lambda k_\beta}^{n\alpha} = \int_{\Delta\Gamma_n}\frac{1}{\beta!}\left[y_0^n - y(\xi,\eta)\right]_{k_1}\Lambda\left[y_0^n - y(\xi,\eta)\right]_{k_\beta} \quad (2\text{-b})$$
$$N^{\alpha}(\xi,\eta)J(\xi,\eta)d\xi d\eta$$

are the moment tensors associated with the sequence multipoles involved in the expansion. In the above expressions, N is number of elements that constitute the boundary, m is number of nodes per element, $N^{\alpha}(\xi,\eta)$ is interpolation function and J (ξ,η) is the Jacobian of the co-ordinate transformation.

5.1. Grouping influence

In the process of evaluating the integrals in large domains, it is found that there are many boundary segments that fall in the far field, and because of the rapid decay of the fundamental solutions, like the Stokeslet and Stresslet, they should be grouped as a unique influence. In the open literature, many approaches have been proposed to do such grouping. The simplest one subdivides the domain into a uniform cube mesh and uses each cell as the base of the group calculations.

A more adaptive way of grouping based on hierarchical trees is called clustering. This procedure uses the same cube mesh, but it is subdivided further only in the places where the boundary is located. This technique generates a recursive quad tree, where each cell (or cluster) has eight children and one parent. In order to add the integrals on the entire boundary segments inside one cell and transfer the values to another we have to be able to add the series without actually evaluating them. Moving the series from centre to centre performs this.

5.2. The first shift

From equation (1) it can be seen that all the moments are evaluated in respect to the point y_0^n, which in our scheme is taken to be the centre of a leaf (the smaller cube in the subdivision). Now the trial solution can be multiplied by the corresponding moments and added together. According with this grouping procedure Equation (1) can be expressed in terms of the following series:

$$\int_{\Gamma} U_{ij}(x,y)\theta_j(y)d\Gamma(y) \sim \sum_{n=1}^{N_l} U_{ij}(x-y_0^n)M_{j0}^{n}$$
$$+\sum_{\beta=1}^{\infty}\sum_{n=1}^{N_l} U_{ij,k_1\Lambda k_\beta}(x-y_0^n)M_{jk_1\Lambda k_\beta}^{n} \quad (3)$$

where N_l represents the total number of leaves and

$$M_{j0}^{n} = \sum_{\alpha=1}^{i}\sum_{\beta=1}^{m}\theta_j^{n\alpha\beta}\mathrm{M}_0^{n\alpha\beta} \quad (4\text{-a})$$

$$M_{jk_1\Lambda k_\beta}^{n} = \sum_{\alpha=1}^{i}\sum_{\beta=1}^{m}\theta_j^{n\alpha\beta}\mathrm{M}_{k_1\Lambda k_\beta}^{n\alpha\beta} \quad (4\text{-b})$$

where i represents the number of elements in the nth leaf and m is the number of nodes per element. From the above equation it is possible to see that further grouping is possible in respect to n. To start the grouping, one must define a new point in respect to which the moments will be grouped. This corresponds to grouping in respect to the centre of a higher-level cluster. Before we proceed further, we have to make clear that the above expansion is not applicable over the whole boundary of the problem, as not the whole surface of the considered domain would belong to the far field for a chosen collocation point. We assume that the part Γ_{nf} of the full boundary Γ belongs to the near field and that the rest belongs to the far-field. The far field boundary is divided into elements which belong to leaves $a_1 \ldots a_s$ which again belong to a cluster on a higher level with centre in point y_1^a, elements which belong to leaves $b_1 \ldots b_w$ which again belong to a cluster with centre in point y_1^b, etc. In this way the above single layer can be can be written as

$$\int_{\Gamma} U_{ij}(x,y)\theta_j(y)d\Gamma(y) = \int_{\Gamma_{nf}} U_{ij}(x,y)\theta_j(y)d\Gamma(y) +$$
$$U_{ij}(x-y_1^a)\sum_{n=a_1}^{a_s}\mu_{j0}^{n} + U_{ij}(x-y_1^b)\sum_{n=b_1}^{b_w}\mu_{j0}^{n} + \Lambda +$$
$$\sum_{\beta=1}^{\infty} U_{ij,k_1\Lambda k_\beta}(x-y_1^a)\sum_{n=a_1}^{a_s}\mu_{jk_1\Lambda k_\beta}^{n} +$$
$$\sum_{\beta=1}^{\infty} U_{ij,k_1\Lambda k_\beta}(x-y_1^b)\sum_{n=b_1}^{b_w}\mu_{jk_1\Lambda k_\beta}^{n} + \Lambda \quad (5)$$

where the new moments are obtained using the following formula

$$\mu^{n}{}_{jk_1\Lambda\, k_\gamma} = M^{n}{}_{jk_1\Lambda\, k_\gamma} + \left(y_1^i - y_0^{i_n}\right)_{k_\gamma} M^{n}{}_{jk_1\Lambda\, k_{\gamma-1}} + \Lambda + \frac{\left(y_1^i - y_0^{i_n}\right)_{k_1} \Lambda \left(y_1^i - y_0^{i_n}\right)_{k_\gamma}}{\gamma!} M^{n}{}_{j0}$$

Now the new moments can be added and the procedure can be continued further grouping the far field influence in larger clusters, yielding

$$\int_\Gamma U_{ij}(x,y)\theta_j(y)d\Gamma(y) = \int_{\Gamma_{nf}} U_{ij}(x,y)\theta_j(y)d\Gamma(y) + \left\{U_{ij}(x-y_p^a)\mu_{j0}^a + U_{ij}(x-y_q^b)\mu_{j0}^b + \Lambda\right\} + \sum_{\beta=1}^{\infty}\left\{U_{ij,k_1\Lambda\, k_\beta}(x-y_p^a)\mu^a_{jk_1\Lambda\, k_\beta} + \right.$$
$$\left. U_{ij,k_1\Lambda\, k_\beta}(x-y_q^b)\mu^b_{jk_1\Lambda\, k_\beta} + \mathrm{K}\right\} \qquad (6)$$

This transformation allows us to move the series from one point to another. In the algorithm this approach is used to move one group of boundary segments to the cell center, and later from cells of a finer level to a coarser one (see figure2).

5.2. The second shift

The series given by any of the first shift around a point y_r^s represents the boundary integral evaluated at the field point x with reference point at y_r^s. For a collocation point x at the boundary which is inside a sphere of radius R with centre at z, chosen here to be the centre of the leaf in which the element with the collocation point is situated, the single layer integral can be expressed as:

$$\int_\Gamma U_{ij}(x,y)t_j(y)d\Gamma(y) \cong \int_{\Gamma_{nf}} U_{ij}(x,y)t_j(y)d\Gamma(y) + \left\{g_{ij0}(z,y_p^a) + g_{ij0}(z,y_q^b) + \Lambda + \right\} \sum_{\beta=1}^{N_s-1}\left\{g_{ijk_1\Lambda\, k_\beta}(z,y_p^a) + g_{ijk_1\Lambda\, k_\beta}(z,y_q^b) + \Lambda\right\}\prod_{i=1}^{\beta}(z-x)_{k_i} \qquad (7)$$

Where

$$g_{ij0}(z,y_r^s) = \mu_{j0}^s U_{ij}(z-y_r^s) + \sum_{\beta=1}^{N_s-1}\mu^s_{jk_1\Lambda\, k_\beta} U_{ij,k_1\Lambda\, k_\beta}(z-y_r^s) \qquad (8\text{-a})$$

$$g_{ijk_1\Lambda\, k_\beta}(z,y_r^s) = \frac{1}{\beta!}\{\mu_{j0}^s U_{ij,k_1\Lambda\, k_\beta}(z-y_r^s) + \sum_{\gamma=1}^{N_s-\beta-1}\mu^s_{jn_1\Lambda\, n_\gamma} U_{ij,n_1\Lambda\, n_\gamma k_1\Lambda\, k_\beta}(z-y_r^s)\} \qquad (8\text{-b})$$

where N_s is the number of terms taken in the original Taylor series. The above expressions is obtained in the same way as before, expanding the integral kernel and its derivatives at a point x around the point ξ and replacing them in the original expansion.

This second shift is done in this way to transfer the far field of a cell to the boundary points inside it by collecting influences, adding the coefficients, and at the end, doing the evaluation at the collocation point on the surface.

To solve the linear system of equations obtained with the present approach, it is necessary to use an indirect solver. This is because the multipole clustering scheme applied to BEM allows us to calculate the value of the boundary integrals relative to a collocation point, but each value of the matrix coefficients cannot be obtained explicitly. In this work, we use the GMRES (k) algorithm (restarted GMRES) with diagonal preconditioning as a numerical solver of the multipole system.

6. PARALLEL DESIGN

The parallel algorithm designed based on multipoles and clusters follows the next guidelines:

6.1. Coarse Parallel and distributed memory

As our principal computational tool is a workstation cluster that is by definition both coarse and distributed. We use MPI as the communication standard for portability since it is supported on most of the hardware available from shared memory supercomputers to networks of PC's and furthermore there is an agreement on the standard that is follow by every implementation.

6.2. No explicit Topology

We do not define explicitly a topology of processors. Clusters of workstations are physically graphs as each machine can communicate with any other; therefore it is possible to assign any symbolic topology on top of it by software (MPI). For most of the communication of multipoles integration we use a simple systolic loop, therefore our model is a closed 1D sequence of processors. In addition to this, there are other instances where communication is performed using MPI collective functions for Broadcast and Collect data that use a different topology model.

6.3. Costzones based on a two-dimensional Grid

A grid $m \times n_1 \times n_2$ of hierarchical trees is used to map the domain. It also represents our Level 0 of refinement. Costzones is a technique designed by Singh et al. [14] that, based on a uniform grid, create a set of contiguous cells that together have a similar workload. The procedure is based on one refinement level of the existing quad tree and uses an estimated load per cell to obtain a recommended workload per processor. This in general provides a more flexible an scalable distribution than other methods like ORB (Orthonormal Recursive Bisection). There are also some restrictions like the number of cells should be bigger than the number of processors to guarantee a well balance system in a highly non-uniform geometry

6.4. Data Replication

Data Replication is used to avoid close field communications and reduce it to only far field. As the grid is used from Level 0, the data transfer between levels of refinement is also local to the processor. Then all the interprocessor communication occurs only at the top of the tree. At this point is when the second shift is important. If we had follow a direct evaluation strategy instead, the communication would have taken place on each of the levels

becoming a non structured transfer, increasing the amount of small communications and making very difficult to model and control.

7. PARALLEL FAST MULTIPOLAR BOUNDARY ELEMENT METHOD

Now we have all what we need to evaluate the integrals of the Stokes hydrodynamic potentials for BEM, using a Fast Multipole Method and Adaptive Clustering. The preliminary steps are as follows: at the beginning the same program is loaded onto each of the processors in the array following a model of SPMD (Single Program Multiple Data) under MPI; each processor loads the geometry that it has been assigned: it includes the real local geometry and the replicated close field; then a hierarchical tree is built for each of the Level 0 cells; the momentums of the segments are calculated for the all the local geometry, as well as the close field integrals.
Each processor performs the following steps for the piece of geometry assigned to it:

1. Coefficients $[M^{n}_{jk_1\Lambda\, k_\beta}]$, for n=1,2,…,p, are calculated once for each boundary segment.
2. Group the coefficients inside each cluster (upward pass): It uses the first shift to move the series from the boundary to the centre of the cluster. Later this cluster moves the result to its parent and so on, until it reaches the Level 0 of refinement. At this moment each cluster of every level has coefficients of a series that represent all the integrals inside itself (see Fig 2).
3. Add the far field of each cluster. On each level, add the influence of all is brothers that are in the far field. This is done accumulating the coefficients in the second shift series. Because the complete tree is not local, part of the far field information is obtained from the other processors by a systolic loop. At this step the communication between processors is necessary.
4.Calculate the final value of the integral on the local nodes (downward pass) by evaluating the power series of the second shift on each node inside every cluster.
5.Collect and ensemble the result. At this point each processor has the value of the far field integral on each of its local nodes and there is only one more step to go add them and build a final result. This step performs a collective communication of the resulting vector.

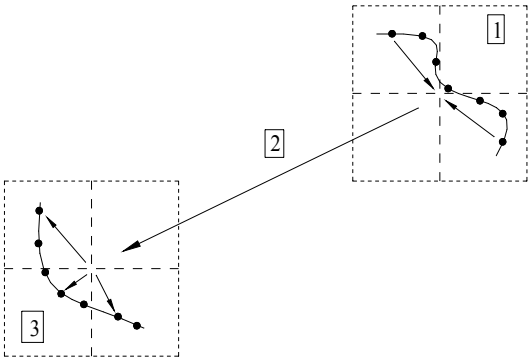


Fig. 2. Two clusters representing three principal steps of the multipolar algorithm: (1) upward pass, (2) transfer of far field and (3) downward pass.

8. CONCLUSION

An application of multipolar technique, parallel programming of indirect BEM formulations for Stokes flow problems is developed. In contrast with the direct formulation, the algebraic systems generated using indirect formulations are stable and have faster convergency with the iterative solvers, without the need of any preconditioning.

By the application of the proposed parallel multipolar-clustering algorithm, good results in memory saving are obtained enabling us to run very large examples prohibitive for traditional BEM implementations. The data structure necessary to solve a problem of 14K nodes was reduced from a potential 10 Gbytes to just over 20 Mbytes, when using eight nodes.

Remarkable improvement in the performance of the BEM solution for Stokes flow was achieved. Although for small problems the clusters-multipolar scheme may be slower than the traditional methods, for large problems it outperforms the former obtaining solutions in a fraction of the traditional BEM.

The structure of multipolar-clustered scheme is naturally parallelisable. There was no need of special designs or lots of additional programming to implement a parallel version of cluster-multipoles. There was some consideration of geometry consistency more to do with the theoretical support rather than the computational limits. As consequence of this implementation, additional reductions were obtained on time spent and in memory requirements, sometimes better than the one to one relation.

9. REFERENCES

[1] H.A. Lorentz, A general theorem concerning the motion of a viscous fluid and a few consequences derived from it, Versl. Kon. Akad Wetensch, Amsterdam 5, 1896, pp.168-174.
[2] G.K. Youngren, A. Acrivos, A Stokes flow past a particle of arbitrary shape: a numerical method of solution, J. Fluid. Mech.1975, pp. 69.
[3] H. Power, G. Miranda, “Second kind integral equation formulation of Stokes flows past a particle of arbitrary shape”, SIAM J. Appl. Math., Vol. 47, 1987, pp. 689-698.
[4] S.J. Karrila, S. Kim, “Integral equations of the second kind for Stokes flow: direct solution for physical variables and removal of inherent accuracy limitations”, Chem. Eng. Commun., Vol. 82, 1989, pp. 123-161.
[5] V. Botte, H. Power, “A second kind integral equation formulation for three-dimensional interior flows at low Reynolds number”, Boundary Elements Communications, Vol. 6, No. 4.
[6] H. Power, “The Low Reynolds Number Deformation of a Gas Bubble in Shear Flow - A General Approach via Integral Equations”, Engineering Analysis with Boundary Elements, Vol. 9, N1, 1991, pp. 31-38
[7] F.J. Briceno, H. Power, “The completed integral equation approach for the numerical solution of the motion of N solid particles in the interior of a deformable viscous drop”, Engineering Analysis with Boundary Elements, 2004, 28(4):333-344.
[8] L. Greengard, M.C. Kropinski, “Integral methods for Stokes flow and isotropic elasticity in the plane”, J. Computational Physics, 125, 1996, pp. 403-414.
[9] J.E. Gómez, H. Power, “A parallel multipolar boundary element method for internal Stokes flows, Intern. J.

Computer Math., Vol. 70, 1999,pp.667-697.
[10] J. Barnes, P. Hut, "A hierarchical O(*N*log*N*) Force Calculation algorithm", Nature, 324, 1986, pp 446-449.
[11] L. Greengard, V. Rokhlin, "A fast algorithm for particle simulations", J. Comp. Phys., 73, 1987, pp. 325-348.
[12] L. Greengard, V. Rokhlin, "A new version of the fast multipole method for the Laplace equation in three dimensions", Acta Numerica, 6, 1997, pp.229.
[13] V. Popov, H. Power, "An O (N) Taylor series multipole boundary element method for three-dimensional elasticity problems", Eng. An. Boundary Elements, 25, 2001, pp.7-18.
[14] J.P. Singh, C. Holt, Totsuka, T., Gupta A, "Hennessy L., Load Balancing and Data Locality in Adaptive Hierarchical N-body Methods: Barnes-Hut, Fast Multipole, and Radiosity", Journal of Parallel and Distribuited Computing June, 1995.

Generalized multivariate rectangular matrix Padé-type approximants and Padé approximants*

Chengde Zheng, Zhibin Li
Department of Mathematics, Dalian Jiaotong University
Dalian, Liaoning 116028, P.R.China
Email: zhengchengde@dl.cn, lizhibin@dl.cn

ABSTRACT

A kind of multivariate matrix Padé-type approximants is studied by the similar ways to those of Brezinski and Kida in scalar cases. With an arbitrary monic bivariate scalar polynomial from the triangular form chosen as the generating one of the approximant, several typical important properties are discussed and the connection between generalized bivariate rectangular matrix Padé-type approximants and Padé approximants is studied. The arguments given in detail in two variables can be extended directly to the case of d variables (d>2).

Keywords: Multivariate Padé approximant, Generalized rectangular matrix Padé-type approximant, Generating polynomial, Covariance.

1. INTRODUCTION

Let us denote by $\mathbf{M}_{r,s}$ the vector space of all rectangular matrices of size $r \times s$ whose entries are real.

Let F be a rectangular matrix formal power series in two real variables u and v:

$$F(u,v) = \sum_{n=0}^{\infty} \sum_{i+j=n} \gamma_{i,j} u^i v^j. \qquad (1)$$

The matrix Padé approximation theory has widely been applied in scattering physics, multiport network synthesis, design of multi-input multi-output digital filters, ARMA model, reduction of a high degree multivariable system, signal processing and systems with the number of inputs different from the number of outputs [1,2,5,6,7].

In view of several potential applications in multivariable two-dimensional (2-D) systems theory, the question of multivariate matrix Padé-type approximants and matrix Padé approximants is urgent to be solved. Bose and Basu [3] studied the existence, non-uniqueness and recursive computation of 2-D matrix Padé approximants with the inverse matrix. By denoting

$$N(u,v) = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \varsigma_{i,j} u^i v^j,\ M(u,v) = \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} \eta_{i,j} u^i v^j.$$

The 2-D matrix Padé approximant is defined as $N(u,v)\{M(u,v)\}^{-1}$ such that

$$F(u,v)M(u,v) - N(u,v) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \varepsilon_{i,j} u^i v^j$$

With $\varepsilon_{i,j} = 0$ for $i = 0,1,\cdots,n_1;\ j = 0,1,\cdots,n_2$ and $i = n_1+1, n_1+2, ..., n_1+m_1+1;\ j = n_2+1, n_2+2, ..., n_2+m_2+1$ excluding the 2-tuple $(i,j) = (n_1+m_1+1, n_2+m_2+1).$ Here all matrices are of size $s \times s\,(r = s).$

Recently Gu Chuanqing [8] considered a bivariate matrix-valued rational interpolant in Thiele-type continued fraction form with scalar denominator and matrix-valued numerator. Similar to the case of vectors, he defined a kind of Samelson-type generalized inverse of matrix A as

$$A^{-1} = \frac{A^T}{\|A\|^2},\ A \neq 0,\ \|A\| = \left(\sum_{i=1}^{p} \sum_{j=1}^{q} a_{i,j}^2 \right)^{\frac{1}{2}},$$

Where $A = (a_{i,j})_{p,q}$, A^T is the transpose matrix of A. So the construction process doesn't need multiplication of matrices.

In our way of approaching this problem, we have preferred to following the method given by Brezinski[4] and Kida[9] in scalar cases: to look for the bivariate rectangular matrix Padé-type approximant on a triangular grid which always exists and is unique when the monic bivariate polynomial $\overline{g}$ of degree q is fixed by a normalization condition. We obtain an order of approximation equal to $p.$ after that we give an expression of the error of approximation. By fixing supplementary conditions, we have increased the order of approximation. A choice of supplementary conditions allows getting a unique monic bivariate polynomial $\overline{g}$ (if it exists). The number of the conditions is linked to r, s and $p, q,$ which requires that some matrix coefficients are partially used. Finally their several typical important properties are discussed and the connection between the bivariate matrix Padé-type approximants and Padé approximants is studied.

As compared to the existing multivariate matrix Padé approximants (cf. [3]), the generalized bivariate rectangular matrix Padé-type approximant doesn't need multiplication of matrices in the construction process. Therefore we don't have to define left-handed and right-handed approximants. It may be useful in the noncommutativity problems of the matrix multiplication. Secondly, the existence condition of the generalized approximant is relaxed if only the generating polynomial is not equal to zero. So it can be applied to singular matrices. Thirdly, the construction of the generalized approximant can be simplified in the compu-

* Supported by the Science and Technology Development Foundation of Education Department of Liaoning Province (2004C060).

tation because it only computes the scalar multiplication instead of usual matrix inverse.

2. NOTATION AND LEMMAS

In order to give the generalized bivariate rectangular matrix Padé-type approximants, we start with regarding $F(u,v)$ as

$$F(u,v)=\gamma_0(u,v)+\gamma_1(u,v)+\ldots+\gamma_n(u,v)+\ldots$$
$$=\sum_{n=0}^{\infty}\gamma_n(u,v), \qquad (2)$$

where $\gamma_n(u,v)$ is a homogeneous rectangular matrix polynomial of degree n in u and v;

$$\gamma_n(u,v)=\sum_{i+j=n}\gamma_{i,j}u^iv^j.$$

Let $\mathbf{H}_{r,s,n}$ be the vector space of all homogeneous rectangular matrix polynomials of degree n in u and v with matrix coefficients belonging to $\mathbf{M}_{r,s}$. Let $\mathbf{G}_n$ be the vector space of all homogeneous scalar polynomials of degree n in u and v with real scalar coefficients. Obviously $\gamma_n(u,v)\in\mathbf{H}_{r,s,n}$.

Let $\mathbf{F}_{r,s}$ denote the vector space of all matrix formal power series $F(u,v)$ as defined in Eq.(2) with matrix coefficients belonging to $\mathbf{M}_{r,s}$. Addition is defined by

$$\sum_{n=0}^{\infty}\alpha_n(u,v)+\sum_{n=0}^{\infty}\beta_n(u,v)=\sum_{n=0}^{\infty}\{\alpha_n(u,v)+\beta_n(u,v)\}.$$

For $r=s$, multiplication is defined by

$$\left(\sum_{n=0}^{\infty}\alpha_n(u,v)\right)\left(\sum_{n=0}^{\infty}\beta_n(u,v)\right)$$
$$=\sum_{n=0}^{\infty}\left(\sum_{i+j=n}\alpha_i(u,v)\beta_j(u,v)\right).$$

Then $\mathbf{F}_{s,s}$ forms a ring. A matrix polynomial in u and v is regarded as an element of $\mathbf{F}_{r,s}$ by natural identification. In Eq. (2), the smallest integer n such that $\gamma_n(u,v)\neq 0$ is called the order of $F(u,v)$ and denoted by $\mathrm{ord}(F)$. The notation $F(u,v)=O(n)$ means that $\mathrm{ord}(F)>n$.

Let $\mathbf{P}$ (respectively $\mathbf{P}_n$) will denote the vector space of all bivariate polynomials (resp. of degree less than or equal to n) whose coefficients are real. Let $\mathbf{P}_{r,s}$ (respectively $\mathbf{P}_{r,s,n}$) denote the vector spaces of all rectangular matrix polynomials (resp. of degree less than or equal to n) in two variables u and v, and whose matrix coefficients belong to $\mathbf{M}_{r,s}$.

Let us consider the "formal Laurent series" in x with coefficients in $\mathbf{P}$; that is, the formal series in x such that the coefficients are polynomials in u and v;

$$h(u,v,x)=a_m(u,v)x^m+a_{m+1}(u,v)x^{m+1}+\cdots,$$
$$a_i(u,v)\in\mathbf{P}, i=m,m+1,\ldots,$$

where m is an integer which may be negative. Addition and multiplication are naturally defined by

$$\sum_i a_i(u,v)x^i+\sum_i c_i(u,v)x^i=\sum_i\{a_i(u,v)+c_i(u,v)\}x^i;$$
$$\sum_i a_i(u,v)x^i\sum_i c_i(u,v)x^i$$
$$=\sum_k\left(\sum_{i+j=k}a_i(u,v)c_j(u,v)\right)x^k.$$

Let $\mathbf{L}$ denote the totality of the above "formal Laurent series." Then $\mathbf{L}$ includes the vector space of all polynomials in x over $\mathbf{P}$. In $\mathbf{L}$, we set

$$\frac{1}{1-x}=1+x+x^2+x^3+\cdots,$$
$$\frac{1}{x^n(1-x)}=x^{-n}+x^{-n+1}+x^{-n+2}+\cdots. \qquad (3)$$

For a given matrix formal power series Eq.(1), let us now define an operator γ by:

$$\gamma\left(\sum_i a_i(u,v)x^i\right)=\sum_i a_i(u,v)\gamma_i(u,v),$$

(With the convention that $\gamma_i(u,v)\equiv 0$ for $i<0$), where the infinite sum $\sum_i a_i(u,v)\gamma_i(u,v)$ is well defined in $\mathbf{F}_{r,s}$ since $\mathrm{ord}(a_i\gamma_i)\geq i$ for $i=0,1,\cdots$.

This operator has the following property:

For any two scalar polynomials $y_1(u,v), y_2(u,v)$ and $h_1(u,v,x), h_2(u,v,x)\in\mathbf{L}$, we have

$$\lambda\left(y_1(u,v)h_1(u,v,x)+y_2(u,v)h_2(u,v,x)\right)$$
$$=y_1(u,v)\lambda\left(h_1(u,v,x)\right)+y_2(u,v)\lambda\left(h_2(u,v,x)\right). \qquad (4)$$

We define the operator $\gamma^{(n)}$ by

$$\gamma^{(n)}\left(h(u,v,x)\right)=\gamma\left(x^nh(u,v,x)\right)$$

For $h(u,v,x)\in\mathbf{L}$, where n is an integer. Then $\gamma^{(n)}$ also has the same property Eq. (4) as γ. Operating γ or $\gamma^{(n)}$ on the special element x^i of $\mathbf{L}$, we have

$$\gamma(x^i)=\gamma_i(u,v) \quad \text{and} \quad \gamma^{(n)}(x^i)=\gamma_{n+i}(u,v),$$

where $\gamma_n(u,v)\equiv 0$ for $n<0$.

We immediately have the following lemma by Eq.(3).

Lemma 2.1. For $n\leq 0$, we have

$$\gamma^{(n)}\left(\tfrac{1}{1-x}\right)=F(u,v).$$

Consider a homogeneous polynomial with respect to the three variables u,v and x which is expressed as follows:

$$\overline{g}(u,v,x)=b_0(u,v)x^q+b_1(u,v)x^{q-1}+\cdots+b_q(u,v),$$
$$b_0(u,v)=1,\ b_i(u,v)\in\mathbf{G}_i. \qquad (5)$$

We call $\overline{g}(u,v,x)$ a monic homogeneous polynomial of degree q. Set

$$g(u,v)=\overline{g}(u,v,1)=b_0(u,v)+b_1(u,v)+\cdots+b_q(u,v),$$

then $g(u,v)\in\mathbf{P}_q$.

Let us now define

$$W^{(n)}(u,v)=\gamma\left(\frac{g(u,v)-x^n\overline{g}(u,v,x)}{1-x}\right),$$

where n is an integer, then $W^{(n)}(u,v)$ is called the n -associated matrix polynomial of $\overline{g}(u,v,x)$ or $g(u,v)$.

About $W^{(n)}(u,v)$, we have the following lemma.

Lemma 2.2. If $W^{(n)}(u,v)$ is defined above, then

$$W^{(n)}(u,v)\in P_{r,s,q+n-1}.$$

Proof.

$$W^{(n)}(u,v)=\gamma\left(\frac{1}{1-x}\left\{\sum_{j=0}^{q}b_j(u,v)-\sum_{j=0}^{q}b_j(u,v)x^{q+n-j}\right\}\right)$$
$$=\sum_{j=0}^{q}b_j(u,v)\gamma\left(\frac{1-x^{q+n-j}}{1-x}\right).$$

If $n>0,$ then

$$W^{(n)}(u,v)=\sum_{j=0}^{q}b_j(u,v)\left\{\gamma_0(u,v)+\ldots+\gamma_{q+n-j-1}(u,v)\right\}.$$

If $n\le 0,$ then

$$W^{(n)}(u,v)=\sum_{j=0}^{q+n-1}b_j(u,v)\gamma\left(1+x+x^2+\ldots+x^{q+n-j-1}\right)$$
$$-\sum_{j=q+n+1}^{q}b_j(u,v)\gamma\left(x^{-1}+x^{-2}+\ldots+x^{q+n-j}\right)$$
$$=\sum_{j=0}^{q+n-1}b_j(u,v)\left\{\gamma_0(u,v)+\ldots+\gamma_{q+n-j-1}(u,v)\right\}.$$

In both cases, since the term $b_j(u,v)\left\{\gamma_0(u,v)+\ldots+\gamma_{q+n-j-1}(u,v)\right\}$ is a matrix polynomial of degree $q+n-1$, we get the result.

Lemma 2.3.

$$F(u,v)g(u,v)-W^{(n)}(u,v)=\gamma^{(n)}\left(\frac{\overline{g}(u,v,x)}{1-x}\right)$$
$$=O(q+n).$$

Proof.

$$F(u,v)g(u,v)-W^{(n)}(u,v)$$
$$=\gamma\left(\frac{1}{1-x}\right)g(u,v)-\gamma\left(\frac{g(u,v)-x^n\overline{g}(u,v,x)}{1-x}\right)$$
$$=\gamma^{(n)}\left(\frac{\overline{g}(u,v,x)}{1-x}\right)=\gamma\left(\sum_{i=0}^{\infty}\sum_{j=0}^{q}b_j(u,v)x^{q-j+n+i}\right)$$
$$=\sum_{i=0}^{\infty}\sum_{j=0}^{q}b_j(u,v)\gamma_{q-j+n+i}(u,v).$$

Noting that $\operatorname{ord}(b_j\gamma_{q-j+n+i})\ge q+n,$ we get the result.

3. GENERALIZED MULTIVARIATE MATRIX PADÉ –TYPE APPROXIMANTS

Firstly we give the following definition:

Definition 3.1. Let $\overline{g}(u,v,x)$ be a monic homogeneous polynomial of degree q and $W(u,v)$ the $(p-q+1)$ -associated matrix polynomial of $\overline{g}(u,v,x)$, i.e.

$$W(u,v)=\gamma\left(\frac{g(u,v)-x^{p-q+1}\overline{g}(u,v,x)}{1-x}\right),$$

where $g(u,v)=\overline{g}(u,v,1).$ Then $W(u,v)/g(u,v)$ is called the generalized bivariate matrix Padé-type approximant and denoted by $(p/q)_F(u,v).$ We call $\overline{g}(u,v,x)$ a generating polynomial of the generalized bivariate matrix Padé-type approximant $(p/q)_F(u,v).$

By Lemmas 2.2 and 2.3 for $n=p-q+1,$ we immediately have:

Theorem 3.2. Let

$$(p/q)_F(u,v)=W(u,v)/g(u,v),$$

where $g(u,v)=\overline{g}(u,v,1).$ Then $g(u,v)$ is a scalar polynomial of degree q and $W(u,v)$ a matrix polynomial of degree $p.$ Moreover,

$$F(u,v)g(u,v)-W(u,v)=\gamma^{(p-q+1)}\left(\frac{\overline{g}(u,v,x)}{1-x}\right)$$
$$=O(p+1). \qquad (6)$$

The generalized bivariate matrix Padé-type approximant generated by $\overline{g}(u,v,x)$ is uniquely determined in the

sense of the following theorem.

Theorem 3.3. Let $(p/q)_F(u,v) = W(u,v)/g(u,v),$ where $g(u,v) = \overline{g}(u,v,1).$ If $\overline{W}(u,v) \in \mathbf{P}_{r,s,p}$ such that

$$F(u,v)g(u,v) - \overline{W}(u,v) = O(p+1), \tag{7}$$

then $\overline{W}(u,v) = W(u,v).$

Proof. It follows from Eq.(6) and Eq.(7) that

$$W(u,v) - \overline{W}(u,v) = O(p+1).$$

As $W(u,v) - \overline{W}(u,v)$ is a matrix polynomial of degree $p,$ we get the result.

Following the proof of Lemma 2.2, we can get

Theorem 3.4. Let $(p/q)_F(u,v) = W(u,v)/g(u,v)$ and

$$g(u,v) = b_0(u,v) + b_1(u,v) + \cdots + b_q(u,v),$$
$$b_0(u,v) = 1, b_i(u,v) \in \mathbf{G}_i.$$

Then $W(u,v)$ is expressed as follows:

$$W(u,v) = \sum_{i=0}^{\min(p,q)} b_i(u,v)\left\{\gamma_0(u,v) + \gamma_1(u,v) + \cdots + \gamma_{p-i}(u,v)\right\}.$$

Theorem 3.5. Let $\overline{g}(u,v,x)$ be a monic homogeneous polynomial of degree $q, g(u,v) = \overline{g}(u,v,1).$ Consider the function

$$\overline{W}(u,v) = \gamma^{(p-q+1)}\left(\frac{g(u,v) - \overline{g}(u,v,x)}{1-x}\right).$$

(a) If $p < q,$ then

$$(p/q)_F(u,v) = \overline{W}(u,v)/g(u,v).$$

(b) If $p \geq q,$ then

$$(p/q)_F(u,v) = \gamma_0(u,v) + \gamma_1(u,v) + \cdots + \gamma_{p-q}(u,v) + \overline{W}(u,v)/g(u,v).$$

Proof. (a) For $p < q,$ we have

$$\gamma^{(p-q+1)}\left(\frac{1}{1-x}\right) = \gamma\left(\frac{1}{1-x}\right) = F(u,v),$$

and

$$\overline{W}(u,v) = \gamma\left(\frac{1}{1-x}\right)g(u,v) - \gamma^{(p-q+1)}\left(\frac{\overline{g}(u,v,x)}{1-x}\right)$$

$$= \gamma\left(\frac{g(u,v) - x^{p-q+1}\overline{g}(u,v,x)}{1-x}\right) = W(u,v).$$

This implies the result of (a).

The result of (b) follows from

$$\left\{\gamma_0(u,v) + \gamma_1(u,v) + \cdots + \gamma_{p-q}(u,v)\right\}g(u,v) + \overline{W}(u,v)$$

$$= \gamma\left(\left(1 + x + \ldots + x^{p-q}\right)g(u,v) + \frac{x^{p-q+1}\left\{g(u,v) - \overline{g}(u,v,x)\right\}}{1-x}\right)$$

$$= \gamma\left(\frac{g(u,v) - x^{p-q+1}\overline{g}(u,v,x)}{1-x}\right) = W(u,v).$$

4. FUNDAMENTAL PROPERTIES

By using Theorems 3.2 and 3.3, we can easily prove the following property:

Property 4.1 (Linearity). Let $Q(u,v) = AF(u,v) + P_k(u,v),$ where $A \in \mathbf{M}_{r,r}, P_k(u,v) \in \mathbf{P}_{r,s,k}, \quad 0 \leq k \leq p-q$ $(P_k(u,v) \equiv 0$ for $p < q).$ Then

$$(p/q)_Q(u,v) = A(p/q)_F(u,v) + P_k(u,v),$$

Provided that the above two approximants have the same bivariate generating polynomial.

From the unicity property, we can easily prove the following result.

Property 4.2 (Translation). If $U_k(u,v) \in \mathbf{H}_{r,s,k},$ $F(u,v) = U_k(u,v)G(u,v),$ then

$$(p+k/q)_F(u,v) = U_k(u,v)(p+k/q)_G(u,v),$$

Provided that the above two approximants have the same generating polynomial.

Proof. If $(p/q)_G(u,v) = W(u,v)/g(u,v),$ then

$$G(u,v)g(u,v) - W(u,v) = O(p+1).$$

On premultiplying both sides of the above equation by $U_k(u,v),$ we obtain:

$$F(u,v)g(u,v) - U_k(u,v)W(u,v) = O(p+k+1).$$

The result follows from Theorem 3.3.

By Property 4.2 and the additivity of generalized bivariate matrix Padé-type approximants, we can easily prove the following property.

Property 4.3. If $F(u,v) = uG(u,v) + vH(u,v);$ then

$$(p+1/q)_F(u,v) = u(p/q)_G(u,v) + v(p/q)_H(u,v),$$

where the above three approximants have the same generating polynomial.

From previous definitions and the construction of $(p/q)_F(u,v),$ we can easily prove the following property.

Property 4.4 (Symmetry). If $F(v,u) = F(u,v),$ i.e. $\gamma_i(v,u) = \gamma_i(u,v)$ $(i = 0,1,2,\ldots),$ then

$$(p/q)_F(v,u) = (p/q)_F(u,v),$$

Provided that the two approximants have the same generating polynomial $g(u,v)$ such that $g(v,u) = g(u,v).$

Denoting $T_1(u) = \mathrm{Proj}_u T(u,v) = T(u,0),$

$$T_2(u) = \mathrm{Proj}_v T(u,v) = T(0,v),$$

where $T(u,v)$ is a scalar polynomial or a matrix polynomial of size $p \times q$. Then we have

Property 4.5. (Projection) Let $(p/q)_F(u,v) = W(u,v)/g(u,v)$, then

$\mathrm{Proj}_u (p/q)_F(u,v) = (p/q)_{F_1}(u) = G(u)/\{\mathrm{Proj}_u g(u,v)\}$,

$\mathrm{Proj}_v (p/q)_F(u,v) = (p/q)_{F_2}(u) = G(u)/\{\mathrm{Proj}_v g(u,v)\}$,

where $G(u), H(v)$ are matrix polynomials of size $p \times q$.

Proof. By the fact that

$$F(u,v)g(u,v) = W(u,v) + O(p+1),$$

We have

$$F_1(u)\mathrm{Proj}_u g(u,v) = \mathrm{Proj}_u W(u,v) + O(u^{p+1}),$$

where $\mathrm{Proj}_u g(u,v)$ is a polynomial of degree q in u and $\mathrm{Proj}_u W(u,v)$ of degree p in $u.$ Therefore the first result follows from the unicity of the univariate matrix Padé-type approximant. The second result can be proved in the similar way.

Property 4.6. Set

$$u' = \frac{au+bv}{1+cu+dv},\ v' = \frac{a'u+b'v}{1+cu+dv}.$$

Let $r = s,\ F(u,v) = G(u',v'),$

$(p/p)_G(u',v') = W(u',v')/g(u',v')$, and

$(p/p)_F(u,v) = Q(u,v)/\{(1+cu+dv)^n g(u',v')\}$;

then $Q(u,v) = (1+cu+dv)^n W(u',v'),$ i.e.

$$(p/q)_F(u,v) = (p/q)_G(u',v').$$

Proof. Firstly we have

$$G(u',v')g(u',v') - W(u',v') = O(n+1).$$

After multiplying both sides by $(1+cu+dv)^n$, we get

$$(p/p)_F(u,v) = \frac{(1+cu+dv)^n W(u',v')}{(1+cu+dv)^n g(u',v')} = (p/p)_G(u',v'),$$

from the unicity property.

From the previous definitions and the unicity property, we can prove the following properties.

Property 4.7. Let

$$(p/q)_F(u,v) = P_1(u,v)/g(u,v)$$

$$= F(u,v) + H_{p+1}(u,v) + O(p+2),\ H_{p+1}(u,v) \in \mathbf{H}_{r,s,p+1},$$

and $(p+1/q)_F(u,v) = P_2(u,v)/g(u,v)$. Then

$$P_2(u,v) = P_1(u,v) - g(0,0)H_{p+1}(u,v).$$

Property 4.8. Set $G(u,v) = AF(u,v)B,$ $A \in \mathbf{M}_{r,r},\ B \in \mathbf{M}_{s,s}.$ If

$$(p/q)_F(u,v) = W(u,v)/g(u,v)$$

and $(p/q)_G(u,v) = Q(u,v)/g(u,v),$

then $Q(u,v) = AW(u,v)B,$ i.e.

$$(p/q)_G(u,v) = A(p/q)_F(u,v)B.$$

5. GENERALIZED MULTIVARIATE MATRIX PADÉ APPROXIMANTS

In this section we want to construct generalized bivariate rectangular matrix Padé-type approximants $(p/q)_F(u,v)$ of order greater than $p.$ In fact we look for a generating polynomial $\overline{g}(u,v,x)$ such that this order is maximal. To obtain that, we have to cancel the matrix coefficients of the first matrix polynomials

$$\gamma^{(p-q+1)}\left(x^i \overline{g}(u,v,x)\right) = \sum_{j=0}^{q} b_j(u,v)\gamma_{p-j+1+i}(u,v).$$

$(i = 0,1,2,...)$ In Theorem 3.2. The unknowns are the coefficients of the unknown polynomials $b_1(u,v),...,b_q(u,v)$ in $\overline{g}(u,v,x)$ which are $(q+3)q/2$ entries. On the other hand, $\gamma^{(p-q+1)}\left(x^i \overline{g}(u,v,x)\right)$ is a homogeneous matrix polynomial of degree $p+i+1$ in u and v. By identically equating the terms of $u^{p+i+1-j}v^j$ $(j = 0,1,...,p+i+1)$ on both sides, we find that every matrix equation group

$$\gamma^{(p-q+1)}\left(x^i \overline{g}(u,v,x)\right) = 0$$

Is equivalent to $p+i+2$ matrix equations of size $r \times s,$ i.e., $(p+i+2)rs$ scalar equations in which appear the $(q+3)q/2$ coefficients of the unknown polynomials $b_1(u,v),...,b_q(u,v)$ linearly.

Let us denote by $E(a)$ the integer part of real number $a.$ Set

$$k = E\left(\sqrt{\left(p+\frac{3}{2}\right)^2 + \frac{(q+3)q}{rs}} - \left(p+\frac{3}{2}\right)\right),$$

$$l = \frac{(q+3)q}{2s} - \frac{(2p+k+3)k}{2}r,$$

then $ls \in \mathbf{Z}$ and $0 \le l < (p+k+2)r.$

For obtaining $(q+3)q/2$ equations with

$(q+3)q/2$ unknowns, we shall have to use k matrix equation groups

$$\gamma^{(p-q+1)}\left(x^i\overline{g}(u,v,x)\right)=0\ (i=0,1,...,k-1)$$

and ls supplementary scalar equations extracted from the following matrix equation group

$$\gamma^{(p-q+1)}\left(x^k\overline{g}(u,v,x)\right)=0 \qquad (8)$$

Which are the first $E(l/r)$ matrix equations of size $r\times s$ and the first $\{l-rE(l/r)\}s$ scalar equations of the $(E(l/r)+1)$-th matrix equation of size $r\times s$. By denoting $t=l-rE(l/r),$ then the above $\{l-rE(l/r)\}s$ scalar equations are the first $E(t)$ lines and the first $\{t-E(t)\}s$ entries of the (E(t) + 1)-th line from the $(E(t)+1)$-th matrix equation of the matrix equation group Eq.(8). Here choosing the first 0 ones indicates choosing none of them.

For simplicity, we denote this choice about the matrix equation group Eq. (8) by

$$\overline{\gamma}^{(p-q+1)}\left(x^k\overline{g}(u,v,x)\right)=0.$$

Theorem 3.2 immediately gives us the following result.

Theorem 5.1. If the monic homogeneous polynomial $\overline{g}(u,v,x)$ of degree q satisfies the following relations

$$\gamma^{(p-q+1)}\left(x^i\overline{g}(u,v,x)\right)=0\ (i=0,1,...,k-1)$$

then the bivariate rectangular matrix Padé-type approximant $(p/q)_F(u,v)$, having $\overline{g}(u,v,x)$ as the generating polynomial, has an order of approximation at least equal to $p+k+1.$

Therefore the generalized bivariate rectangular matrix Padé-type approximant $(p/q)_F(u,v),$ having $\overline{g}(u,v,x)$ as the generating polynomial is an approximant of $F(u,v)$ with an order at least equal to $p+k+1.$ Moreover some lines and entries have an order of approximation at least equal to $p+k+2$ with respect to the corresponding lines and entries of $F(u,v)$. This approximant will be called a generalized bivariate rectangular matrix Padé approximant of $F(u,v)$.

Now we study the problem of existence and uniqueness of polynomial $\overline{g}(u,v,x)$.

Let us write

$$b_i(u,v)=\sum_{j=0}^{i}b_{i,j}u^{i-j}v^j,\ b_{i,j}\in\mathbf{R}$$

In Eq.(5). From above discussion, $\overline{g}(u,v,x)$ exactly satisfying the following relations

$$\begin{cases}\sum_{i=0}^{q}\sum_{j=0}^{i}\gamma_{p+1-i}(u,v)b_{i,j}u^{i-j}v^j=0,\\ \sum_{i=0}^{q}\sum_{j=0}^{i}\gamma_{p+2-i}(u,v)b_{i,j}u^{i-j}v^j=0,\\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots\\ \sum_{i=0}^{q}\sum_{j=0}^{i}\gamma_{p+k-i}(u,v)b_{i,j}u^{i-j}v^j=0,\\ \sum_{i=0}^{q}\sum_{j=0}^{i}\overline{\gamma}^{(p-q+1)}\left(x^{p+k-i}\right)b_{i,j}u^{i-j}v^j=0.\end{cases} \qquad (9)$$

By setting to zero the coefficients of the terms of $u^{p+m+1-i}v^i\ (m=0,1,...,k;\ i=0,1,...,p+m+1)$ on the left sides of every matrix equation group in system Eq.(9), we get a system (*) including $(q+3)q/2$ scalar equations in which appear the coefficients of the unknown polynomials $b_i(u,v)(i=1,2,...,q)$ linearly. Let us denote by $M^{(p,q)}$ the block coefficient matrix of system (*), then

$$M^{(p,q)}\in\mathbf{M}_{q(q+3)/2,q(q+3)/2}.$$

We obviously obtain the following result.

Theorem 5.2. The bivariate rectangular matrix Padé approxi- mant $[p/q]_F(u,v)$ exists and is unique if and only if the coefficient matrix $M^{(p,q)}$ of the system (*) is regular.

6. REFERENCES

[1] A.I. Aptekarev, W.V. Assche, "Scalar and matrix Riemann-Hilbert approach to the strong asymptotics of Padé approximants and complex orthogonal polynomials with varying weight," Journal of Approximation Theory, 129(2004), 129-166.

[2] Baker,P.R.Graves-Mories, Padé Approximants, Part II, Addison-wesley Publishing Company, 1981.

[3] N.K. Bose, S. Basu, "Two-dimensional matrix Padé approximants: existence, nonuniqueness and recursive computation," IEEE Transactions on Automatic Control, 20(1980), 509-514.

[4] C. Brezinski, Padé-Type Approximation and General Orthogonal Polynomial, ISNM 50, Basel: Birkhäuser, 1980.

[5] J.R. Cardoso, F.S. Leite, "Padé and Gregory error estimates for the logarithm of block triangular matrices," Applied Numerical Mathematics, 56(2006), 253-267.

[6] C.F.Chen, "Model reduction of multivariable control systems by means of a matrix continiued fraction," International Journal of Control, 20(1974), 225-238.

[7] C.F.Chen, L.S.Shieh, "A novel approach to linear model simplification," International Journal of Control, 8(1968), 561- 570.

[8] C.Q.Gu, "Bivariate Thiele-type matrix-valued rational interpolants," Journal of Computational and Applied Mathematics, 80(1997), 71-82.

[9] S. Kida, "Padé-type and Padé approximants in several variables," Applied Numerical Mathematics, 6(1989/90), 371-391.

A Parallel Algorithm for Longest Common Substring of Multiple Biosequences *

Wei Liu
Department of Computer Science, Yangzhou University
Yangzhou 225009 China
Email: yzliuwei@126.com
Ling Chen
National Key Lab of Novel Software Tech, Nanjing University
Nanjing 210093 China
Email: lchen@yzcn.net

ABSTRACT

A parallel algorithm *FAST_LCS* for the longest common substring (LCS) problem is presented. For two sequences X and Y with length n and m, the time complexity of parallel computing is O (|LCS(X, Y)|), where |LCS(X, Y)| is the length of the LCS of X, Y. For n biosequences $X_1, X_2, \ldots, X_n$, time complexity of our parallel algorithm is $O(|LCS(X_1, X_2, \ldots, X_n)|)$ which is independent of the number of sequences n. Experimental result on the gene sequences of *tigr* database shows that our algorithm is faster and more efficient than other LCS algorithms and can get exactly correct result.

Key words: bioinformatics; longest common subsequence.

1. INTRODUCTION

Searching for the longest common subsequence (LCS) of biosequences is one of the most important problems in bioinformatics [1,2,3,4,5]. Presented in 1981, Smith-Waterman algorithm [6] is a well known LCS algorithm which was evolved from the Needleman-Wunsch [7] algorithm for LCS of two sequences. For the LCS problem of multiple sequences [8,9], time complexity grows very fast when the number of the sequences increases. The MSA program [10] can process up to ten closely related sequences. Stoye described a new divide and conquer algorithm DCA [11] sits on top of MSA and expands its capabilities. Clustal-W [12] is one of the most widely used multiple sequence alignment software. It was developed by improving Feng and Doolittle's algorithm [13] so as to enhance the correctness. Although the improvement is done, strong limitations remain on the number of sequences that can be handled.

In this paper, we present a fast algorithm named FAST_LCS for the LCS of multiple biosequences. The algorithm first seeks the successors of the initial identical character tuples according to a successor table to obtain all the identical tuples and their levels. Then by tracing back from the identical character tuple with the largest level, the result of LCS can be obtained. For n sequences $X_1, X_2, \ldots, X_n$, time complexity of sequentially execution of our algorithm is O(L), here L is the number of identical character tupelos, and time complexity of our parallel algorithm is $O(|LCS(X_1, X_2, \ldots, X_n)|)$ which is independent of the number of sequences n. Experimental result on the gene sequences of *tigr*[14] database using MPP parallel computer Shenteng 1800 shows that our algorithm is faster and more efficient than other LCS algorithms and can get exactly correct result.

2. IDENTICAL CHARACTER PAIR AND ITS SUCCESSOR TABLE

Let $X=(x_1, x_2, \ldots, x_n)$, $Y=(y_1, y_2, \ldots, y_m)$ be two biosequences, where $x_i, y_i \in \{A,C,G,T\}$. We can define an array CH of the four characters as A=CH (1), C=CH (2), G=CH (3) and T=CH (4).To find their longest common subsequence, we first build successor tables of the identical characters for these two strings. The successor tables of the identical characters of X and Y are denoted as TX and TY which are 4*(n+1) and 4*(m+1) two dimensional arrays. $TX(i, j)$ is defined as follows.

Definition1. For the sequence $X=(x_1, x_2, \ldots, x_n)$, its successor table TX of identical character is defined as :

$$TX(i,j)=\begin{cases}\min\{k \mid k \in SX(i,j)\} & SX(i,j)\neq\phi \\ - & \text{otherwise}\end{cases} \quad \text{Eq. (1)}$$

Here, $SX(i, j)=\{k| \ x_k =CH(i), k>j\}$, i = 1,2,3,4, j = 0,1,…n. It can be seen from the definition that if $TX(i, j)$ is not "—", it indicates the position of the next character identical to CH(i) after the x_j in sequence X, if $TX(i, j)$ is equal to "—", it means there is no character CH(i) after x_j.

Example 1 Let X = " T G C A T A", Y = " A T C T G A T". Their successor tables TX and TY are:

TX:

i	0	1	2	3	4	5	6
1	4	4	4	4	6	6	-
2	3	3	3	-	-	-	-
3	2	2	-	-	-	-	-
4	1	5	5	5	5	-	-

TY:

i	0	1	2	3	4	5	6	7
1	1	6	6	6	6	6	-	-
2	3	3	3	-	-	-	-	-
3	5	5	5	5	5	-	-	-
4	2	2	4	4	7	7	7	-

Definition2. For the sequences X and Y, if $x_i = y_j$, we call them an identical character pair of X and Y, and denote it as (i, j). The set of all the identical character pairs of X and Y is denoted as $S(X, Y)$.

Definition3. Let (i, j) and (k, l) be two identical character pairs of X and Y. If $i<k$ and $j<l$, we call (i, j) a predecessor of (k, l), or (k, l) a successor of (i, j), and denote them as $(i, j)<(k, l)$.

Definition4. Let $P(i, j) = \{(r, s)|(i, j) < (r, s), (r, s) \in S(X, Y)\}$ be the set of all the successors of identical pair (i, j), if $(k, l) \in P(i, j)$ and there is no $(k', l') \in P(i, j)$ satisfying the condition: $(k', l') < (k, l)$, we call (k, l) a direct successor of (i, j), and denoted it as $(i, j) \prec (k, l)$.

Definition5. If identical pair $(i, j) \in S(X, Y)$ and there is no $(k, l) \in S(X, Y)$ so that $(k, l) < (i, j)$, we call (i, j) an initial identical pair.

Definition6. For an identical pair $(i, j) \in S(X, Y)$, its level is defined as follows:

$$level(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ is an initial identical character pair} \\ \max\{level(k,l)+1 \mid (k,l) < (i,j)\} & \text{otherwise} \end{cases} \quad \text{Eq. (2)}$$

From the definitions above, the following lemma can be easily deduced:

Lemma1. Denote the length of the longest common subsequence of X, Y as $|LCS(X, Y)|$, then $|LCS(X, Y)| = \max\{\text{level}(i, j) \mid (i, j) \in S(X, Y)\}$.

3. THE OPERATIONS OF PRODUCING SUCCESSORS AND PRUNING

In the first step of our algorithm, all direct successors of all the initial identical character pairs can be produced using the successor tables. Then, the direct successors of all those successors produced in the first step are generated. Repeat these operations of generating the direct successors until no more successors could be produced. Therefore, producing all the direct successors for the identical character pairs is a basic operation in our algorithm.

For an identical character pair $(i, j) \in S(X, Y)$, the operation of producing all its direct successors is as follows:

$$(i,j) \to \{(TX(k,i), TY(k,j)) \mid k = 1,2,3,4, TX(k,i) \neq '-' \text{ and } TY(k,j) \neq '-'\} \quad \text{Eq. (3)}$$

From (3) we can see that this operation is to couple the elements of the ith column of TX and the jth column of TY to get the pairs. For instance, the operation on the identical character pair (2,5) in Example 1 is illustrated as follows:

$$(2\ \ 5) \to \begin{bmatrix} 4 & 6 \\ 3 & - \\ - & - \\ 5 & 7 \end{bmatrix} \to \left\{ \begin{pmatrix} 4 & 6 \end{pmatrix} \begin{pmatrix} 3 & - \end{pmatrix} \begin{pmatrix} - & - \end{pmatrix} \begin{pmatrix} 5 & 7 \end{pmatrix} \right\} \to \left\{ \begin{pmatrix} 4 & 6 \end{pmatrix} \begin{pmatrix} 5 & 7 \end{pmatrix} \right\}$$

Since $(3,-)$ and $(-,-)$ do not represent identical character pairs, they only indicate the end of the process of searching for the successors in this branch. After discarding $(3,-)$ and $(-,-)$, the successors of (2,5) are just (4,6) and (5,7). It should be point out that the successors produced in the operation are not all direct successors of (i, j). For example, (5,7) is not the direct successor of (2,5), since $(2, 5) \prec (4, 6) \prec (5, 7)$.

Lemma2. For an identical character pair (i, j), the method illustrated above can produce all its successors.

Proof of Lemma 2 is omitted due to the limited space. It is obvious that $(TX(k, 0), TY(k, 0))$, k=1, 2, 3, 4, are all the initial identical pairs of X and Y. By Lemma 2, we know that starting from those initial identical pairs, all the identical pairs and their levels can be produced by the method illustrated above. In such process of generating the successors, prune technique can be implemented to remove the identical pairs which can not generate the longest common subsequence so as to reduce the searching space and accelerate the speed of process.

Theorem1. If on the same level, there are two identical character pairs (i, j) and (k, l) satisfying $(k, l)>(i, j)$, then (k, l) can be pruned without affecting the algorithm to get the longest common subsequence of X and Y.

Proof of Theorem 1 is omitted due to the limited space. By Theorem 1, the pruning process can be implemented to remove all those redundant identical pairs. For instance, (4, 6) and (5, 7) in Example 1 are the successors of the identical pair (2, 5). Since they are on the same level and $(4, 6) \prec (5, 7)$, we can prune (5, 7) by Theorem 1.

Some other prune operations are also helpful to reduce the searching space. These prune operations are based on the following theorems and corollaries. Due to the limited space, we omit the proofs for these theorems and corollaries.

Theorem2. If on the same level, there are two identical character pairs (i_1, j) and (i_2, j) satisfying $i_1<i_2$, then (i_2, j) can be pruned without affecting the algorithm to get the longest common subsequence of X and Y.

Corollary1. If there are identical character pairs (i_1, j), (i_2, j), $\cdots$, (i_r, j) on the same level and $i_1<i_2<\cdots<i_r$, then we can prune $(i_2, j), \cdots, (i_r, j)$.

4. FRAMEWORK OF THE ALGORITHM

Based on the operations of generating the successors of the identical character pairs using successor tables and the pruning technology, we present a fast longest common subsequence algorithm FAST_LCS. The algorithm consists of two phases: the phase of searching for all the identical character pairs and the phase of tracing back to get the longest common subsequences. The first phase begins with the initial identical character pairs, then continuously searches for their successors using the successor tables. In this phase, the pruning technology is implemented to discard those search branches which obviously can't obtain the optimum solution so as to reduce the search space and speed up the process of searching. In the algorithm, a table called *pairs* is used to store the identical character pairs obtained in the algorithm. In the table *pairs*, each record takes the form of (*k*, *i*, *j*, *level*, *pre*, *state*) where the data items denote the index of the record, the identical character pair (i,j), its level, index of its direct predecessor and its current state. Each record in *pairs* has two states. For the identical pairs whose successors have not been searched, it is in *active* state; otherwise it is in *inactive* state. In every step of search process, the algorithm searches for the successors of all the identical

pairs in *active* state in parallel. Repeat this search process until there is no identical pair in *active* state in the table. The phase of tracing back starts from the identical pairs with the maximal level in the table, and traces back according to the *pred* of each identical pair. If there are more than one identical pair with the maximal level in the table, the tracing back procedure for those identical pairs can be carried out in parallel and several longest common subsequences can be obtained concurrently. This tracing back process ends when it reaches an initial identical pair, and the trail indicates the longest common subsequence. The framework of the algorithm FAST_LCS is as follows:

Algorithm-FAST_LCS (*X*,*Y*)

Input *X* and *Y*: Sequences with lengths of *m* and *n* respectively;

Output LCS : The longest common subsequence of *X*,*Y*;

Begin

1. Build tables *TX* and *TY*;
2. Find all the initial identical character pairs: (*TX*(*k*, 0),*TY*(*k*, 0)), *k*=1,2,3,4;
3. Add the records of the initial identical pairs (*k*,*TX*(*k*, 0),*TY*(*k*, 0), 1, ϕ, *active*), *k*=1,2,3,4 to the table *pairs*.
   /* For all the initial identical pairs, their *level*=1, *pre*= ϕ and *state=active**/
4. Repeat
   For all *active* identical pairs (*k*, *i*, *j*, *level*, *pre*, *active*) in *pairs* parallel do
   Produce all the successors of (*k*, *i*, *j*, *level*, *pre*, *active*).
   For each identical character pair (*g*, *h*) in the successor set of (*k*, *i*, *j*, *level*, *pre*, *active*), a new record (*k'*, *g*, *h*, *level*+1, *k*, *active*) is generated and inserted into the table *pairs*.
   Change the state of (*k*, *i*, *j*, *level*, *pre*, *active*) into *inactive*.
   End for
   Use prune operation on all the successors produced in this level to remove all the redundant identical pairs from table *pairs*.
5. Until there is no record in *active* state in table *pairs*.
6. Compute *r*= the maximal level in the table pairs.
7. For all identical pairs (*k*, *i*, *j*, *r*, *l*, *inactive*) in *pairs* parallel do
   pred = *l*; *LCS*(*r*) = x_i.
   While *pred* $\neq \phi$ do
   7.1.1 get the *pred*-th record (*pred*, *g*, *h*, *r'*, *l'*, *inactive*) from table *pairs*.
   7.1.2 *pred* = *l'*; *LCS*(*r'*) = x_g.
   end while

7.4 End for

End.

Assume that the number of the identical character pairs of *X*, *Y* is *L*. In our algorithm, every identical pair must have the operation of producing successors at least once. Because of the pruning technology, this operation can only be implemented exactly once on each identical character pair. Therefore the time complexity for sequentially executing of the algorithm FAST_LCS (*X*, *Y*) is *O* (*L*). Since the table *pairs* has to store all the identical character pairs, it requires *O* (*L*) memory space. Considering that the memory space cost of *TX*, *TY* are 4*(*n*+1) and 4*(*m*+1), the storage complexity of our algorithm is max {4*(*n*+1) +4*(*m*+1), *L*}. In parallel computing, since the computation for each identical pair can be assigned on one processor, the process on all the identical pairs can be carried out in parallel. Therefore, process of each level requires *O* (1) time, and the time required for the parallel computation is equal to the maximal level of the identical pairs. By Lemma 1, we know that the length of the longest common subsequence of *X*, *Y*, |LCS (*X*, *Y*)|, is equal to the largest level of the longest common subsequence of the identical pairs. Therefore the time complexity for parallel executing of FAST_LCS is *O* (|LCS(*X*, *Y*)|).

5. EXTEND TO THE LCS OF MULTIPLE SEQUENCES

Our algorithm FAST_LCS (*X*, *Y*) can be easily extended to the LCS problem of multiple sequences. Suppose there are *n* sequences $X_1, X_2, \ldots, X_n$, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{i,n_i})$, n_i is the length of X_i, $x_{ij} \in \{A,C,G,T\}$. To find their longest common subsequence, similar to the case of two sequences, the algorithm for multiple sequences first builds the successor tables for all the sequences. Denote the successor tables for X_1, X_2, …, X_n as TX_1, TX_2, … , TX_n, each of which is a two-dimensional array of size 4*(n_i+1). Similar to identical character pair in the case of two sequences LCS, we define the concept of identical character tuple in the algorithm for LCS of multiple sequences:

Definition7. For the sequences $X_1, X_2, \ldots, X_n$, if $x_{1,i_1} = x_{2,i_2} = \ldots = x_{n,i_n}$, we call them an identical character tuple of the sequences $X_1, X_2, \ldots, X_n$ and denote it as $(i_1, i_2, \ldots, i_n)$.

Similar to the case of two sequences LCS, all direct successors of all the initial identical character tuples are produced using the successor tables. Then, the direct successors of all those successors produced in the first step are generated. Repeat these operations of generating the direct successors until no more successors could be produced. In such process of generating the successors, prune technique can be implemented to remove the identical tuples which can not generate the longest common subsequence so as to reduce the searching space and accelerate the speed of process. All the theorems and corollaries of pruning technique for two sequences LCS can be easily extended to the case of multiple sequences.

Let the number of the identical character tuples of the sequences $X_1, X_2 \ldots X_n$ be *L*. In our algorithm, since every identical tuple must have the operation of producing successors exactly once, the time complexity for sequentially executing of the algorithm on the sequences $X_1, X_2 \ldots X_n$ is *O* (*L*) which is independent of the number of the sequences *n*. This means our algorithm is much more efficient for the LCS problem of large number of sequences. In parallel implementation, all the process on the identical tuples can be carried out in parallel. Therefore, process of each level requires *O* (1)

time, and time complexity of parallel execution of our algorithm is O (|LCS(X_1, X_2… X_n)|).

It should be point out that in most of the algorithms for LCS of multiple sequences, time complexity strongly depends on the number of the sequences. The time complexity of our algorithm is O (L) for sequential computation and O (|LCS(X_1, X_2… X_n)|) for parallel implementation which are independent of the number of sequences n. This means our algorithm is much more efficient for the LCS problems of large number of sequences.

6. EXPERIMENTAL RESULTS

6.1 The results of sequential execution of the algorithm

We test our algorithm FAST_LCS on the multiple sequences of *tigr*[14] data base and compare it with the Clustal-W algorithm which is the most common used algorithm for multiple sequences. Fig.1 and Fig.2 show the comparison of the computation time of our algorithm FAST_LCS and that of Clustal-W[12] algorithm.

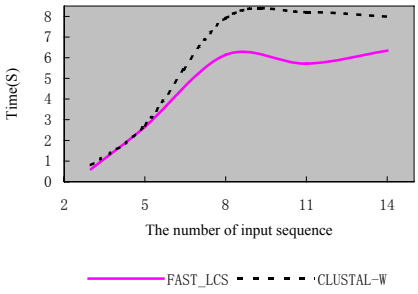


Fig.1. Comparison of the computation time of FAST_LCS with that of Clustal-W on sequences sets of different numbers of sequences

From Fig.1, we can see that FAST_LCS is faster than Clustal-W for sets with different numbers of sequences. Especially when the number of the sequences is larger than 5, FAST_LCS takes even much less time than Clustal-W.

From Fig.2, we can see that FAST_LCS is faster than Clustal-W for sequences sets with different lengths.

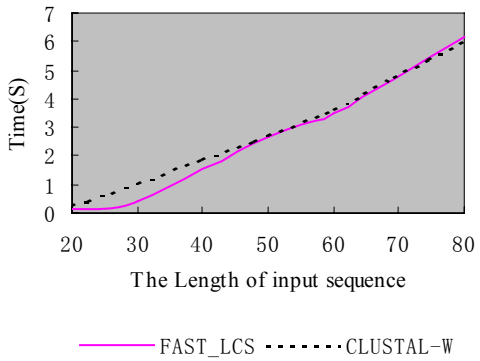


Fig. 2. Comparison of the computation time of FAST_LCS with that of Clustal-W on sequences sets of different lengths

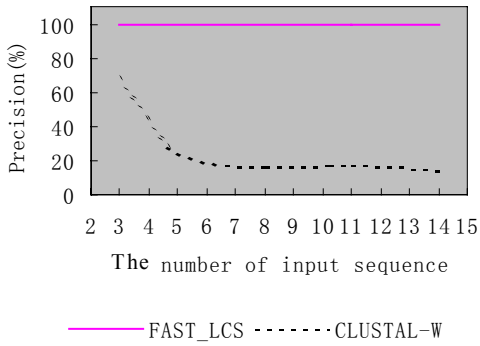


Fig. 3. Comparison of the precision of FAST_LCS with that of Clustal-W on sets on sets with different numbers of sequences

We also compare the precision of our FAST_LCS algorithm with that of Clustal-W algorithm. Here precision is defined as:

$$\text{Precision} = \frac{\text{The length of the common subsequence computed by the algorithm}}{\text{The length of the longest common subsequence in correct match}}$$

Fig.3 shows the comparison of precision of FAST_LCS with that of Clustal-W on the sets with different numbers of sequences, while Fig.4 shows the comparison of precisions of the two algorithms on the sets of sequences with different lengths. From these two figures, we can see that no matter how the length and the number of sequences are increased, our algorithm can obtain precise results. The precision of Clustal-W declines when the number or the length of the sequences is increased. Therefore the precision of our algorithm is much higher than Clustal-W algorithm.

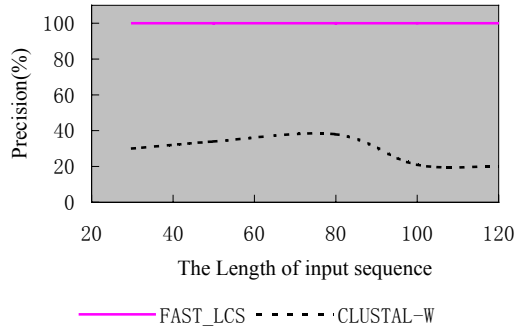


Fig. 4. Comparison of the precision of FAST_LCS with that of Clustal-W on sequences sets of different lengths

6.2 The results of parallel computing

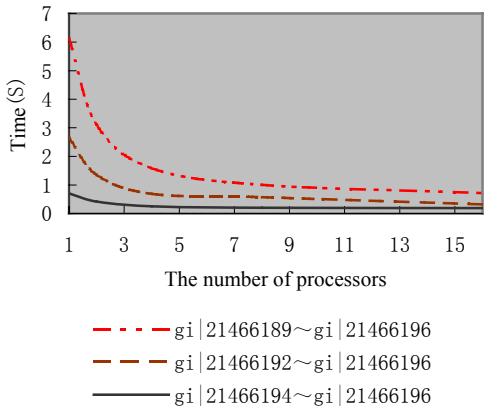


Fig. 5. Parallel computational time of FAST_LCS using different processor numbers

We also test our algorithm on the rice gene sequence from *tigr* database on the massive parallel processors Shenteng 1800 using MPI (C bounding). The experimental results by using different numbers of processors are shown in Fig.5. Three sets of gene sequences are tested. The names, the number of input sequence and computation times are listed in Table 1. From Fig.5 and Table 1 we can see that the computation speed will become faster as the number of processors increases. But because of the overhead of communication between processors which increases the total time of the algorithm, the speedup of our algorithm can not increase linearly with the increasing of processors exactly. This is in conformity with the Amdahl's Law.

Table 1. Computational time of parallel FAST_LCS using different numbers of processors

Sequence name	The number of sequence	Computational time using different numbers of processors (s)				
		1	2	4	8	6
gi|21466189 ～ gi|21466196	8	6.14	3.10 93	1.58 49	1.00 63	0.71 54
gi|21466192 ～ gi|21466196	5	2.65 6	1.35 79	0.69 21	0.57 07	0.31 73
gi|21466194 ～ gi|21466196	3	0.70 9	0.42 14	0.25 382	0.20 445	0.19 042

7. CONCLUSION

On the premise of guaranteeing precision of the results of LCS, we present a fast algorithm named FAST_LCS for the longest common substring (LCS) problem. The algorithm first seeks the successors of the initial identical character tuples according to a successor table to obtain all the identical tuples and their levels. Then by tracing back from the identical character tuple with the largest level, the result of LCS can be obtained. For n biosequences X_1, X_2, … , X_n, time complexity of sequentially execution of our algorithm is O(L), here L is the number of identical character tuples, and time complexity of our parallel algorithm is $O(|LCS(X_1, X_2, \ldots, X_n)|)$ which is independent of the number of sequences n. Experimental result on the gene sequences of *tigr* database using MPP parallel computer Shenteng 1800 shows that our algorithm can get exactly correct result and is faster and more efficient than other LCS algorithms.

REFERENCES

[1] B.L.Hao, S.Y.Zhang, The manual of Bioinformatics [M].Shanghai science and technology publishing company，2000,171-172.

[2] Y.D.Li, Z.R.Sun, et al, Bioinformatics-The practice guide for the analysis of gene and protein. Tsinghua university publishing company, 2000,138.

[3] E.W. Edmiston, N.G.Core, et al, "Parallel processing of biological sequence comparison algorithms". International Journal of Parallel Programming, 17(3), 1988,259-275.

[4] E.Lander , "Protein sequence comparison on a data parallel computer. In": Proceedings of the 1988 International Conference on Parallel Processing, 1988,257-263.

[5] A.R. Galper, D.L.Brutlag, "Parallel similarity search and alignment with the dynamic programming method. Technical Report", California: Stanford University, 1990.

[6] T.F.Smith, M.S.Waterman, "Identification of common molecular subsequence". Journal of Molecular Biology, 1990,215,403-410.

[7] S.B.Needleman, C.D.Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins". Journal of Molecular Biology, 48(3), 1970,443-453.

[8] H. Carrillo, D.J.Lipman, "The multiple sequence alignment problem in biology". SIAM Journal Application, Math. 48, 1073-1082 (1988).

[9] K. Reinert, J. Stoye, et al, "An iterative method for faster sum-of-tuple multiple sequence alignment. Bioinformatics 16(9)", 2000,808-814.

[10] D.J.Lipman, S.F.Altschul, et al, "A tool for multiple sequence alignment. Proc.Natl". Acad. Sci. USA 86, 1989,4412-4415.

[11] J. Stoye, V. Moulton, et al, "DCA: an efficient implementation of the divide-andconquer approach to simultaneous multiple sequence alignment". Compute Application, Biosci.13 (6), 1997,625-626.

[12] J.D. Thompson, D.G. Higgins, et al, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice". Nucleic Acids Research, 1994,22(22), 4673-4680.

[13] D.F.Feng, R.F.Doolittle, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees". Journal of Molecular Biology, 1987, pp.351-360.

[14] http://www.tigr.org/tdb/benchmark.

The BFGS Parallel Algorithm of Unconstrained Optimization Problems In Distributed Environment

Wenjing Li, Binglian Chen
Department of Information Technology, Guangxi Teachers Education University, Nanning, 530001,China
Email:liwj@gxtc.edu.cn
Qingping Guo
Department of Computer Science, Wuhan University of Technology, Wuhan, 430063, China
Email: qpguo@mail.whut.edu.cn

ABSTRACT:

Based on analysis of both the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm and its convergence properties of unconstrained optimization problems, this paper presents a parallel algorithms of BFGS by using the row interleaved format parallel decomposition of CHOLESKY, the synchronous parallel Wolfe-Powell non-linear search and the modified formula of BFGS. The paper also analyses the time complexity and accelerating rate of the algorithm. The experimental results of PC cluster show that the theoretical analysis and the experimental results of BFGS parallel algorithm are consistency with a marked acceleration rate.

Keywords: distributed platforms; unconstrained optimization; BFGS parallel algorithm; complexity.

1. INTRODUCTION

The optimization problem can be found universally in industrial and agricultural production, engineering and technology, transportation, production management, economic planning, defense, finance, and other fields. Many practical problems can be transferred into abstract optimization problem. For the equations of optimization, a lot of people structured many classic algorithms, and for some smaller scale issues, these algorithms have made the sequential realization of most of them. However, in non-linear equations such as optimization methods used in the predictability research and the sensitivity analysis in weather and climate, petroleum, geological surveying, and other large-scale problems, the processing time of current sequential algorithms has exceeded people's endurance and the solving of the practical problems has no significance. Therefore, on the BFGS sequential algorithm basis, this article proposes and implements a parallel BFGS algorithms, analyzing its complexity, speedup rate and other performances of parallel algorithm. The experimental results of PC cluster shows that the theoretical analysis of this algorithm and the experimental results are consistent.

2. The definition of unconstrained optimization problems and the BFGS algorithm

1.1The basic concept of unconstrained optimization problems

The functions f, g_i, h_j: $R^n \to R$, i=1, …, m_1, $j=m_{1+1}$, …, m_{1+p} are assumed to be twice continuously differentiable. Constrained optimization problems are

$$\min f(x) \tag{1.1}$$

$$s \cdot t \cdot\ g_i(x) \geq 0,\ i \in \mathrm{I}=\{1, \cdots, m_1\} \tag{1.2}$$

$$h_j(x) = 0,\ j \in E=\{m_{1+1},\ldots,m_{1+p}\} \tag{1.3}$$

The question feasible territory is

$$D=\{ x \in R^n \mid g_i(x) \geq 0,\ i \in \mathrm{I},\ h_j(x) = 0,\ j \in \mathrm{E}\}$$

Among $x = (x_1, x_2, \cdots, x_n) \in R^n$ is the n dimension vector. f, $g_i(x)$, $h_j(x)$, i=1, … m is x function; *s. t.* stands for "subject to", expressing x receiving the constrained condition of (1. 2), (1. 3); $f(x)$ is called the objective function; (1.2) is called the inequality constrained; (1.3) is called the equality constrained. If. $m_{l+p} = 0$, then constrained optimization problems draws back to become unconstrained optimization problems [1].

Unconstrained optimization problems algorithm includes the descent method, the steepest descent method, Newton's method, Quasi-Newton's method, the conjugate gradient method, the direct method and the trust-region algorithm and so on. Among them, using BFGS (Broyden–Fletcher-Goldfarb-Shanno) the correction formula to solve unconstrained non-linear optimization problems is called the BFGS algorithm. It is one of Quasi-Newton's methods, which has the following three characteristics[2]:

(1) Select some approximate matrix B_k to substitute $\Delta^2 f(x)$; $\Delta^2 f(x)$ expresses Hessian matrix of $f(x)$ in x_k point, causing the direction of the corresponding algorithm approximate with Newton direction. This ensures the algorithm has the quick convergent rates and possesses super linear convergence properties under certain conditions.

(2) To all k, matrix B_k is symmetrical positive definite, thus causing the direction which the algorithm produces is the descending direction of function $f(x)$ in x_k point;(3) Matrix. B_k is easy to calculate.

2.2 BFGS algorithm

The sequential BFGS algorithm step is as follows:

Step 1. Take the initial point $x^{(0)} \in R^n$, initial symmetrical positive definite matrix. $B_0 \in R^{n \times n}$, precision ε >0; make *K*=0;

Step 2. If norm $\| \Delta f(x)^{(k)} \| \leq \varepsilon$, the algorithm should be terminated, resulting in the solution $x^{(k)}$, otherwise, move

to the third step;
Step 3. Solve the linear system of equations

$$B_k d + \nabla f(x^{(k)}) = 0 \quad (1.4)$$

The solution is $d^{(k)}$.

Step 4. By linear search steplength α_k.

Step5. Make k=k+1 and compute $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$. If norm $\|\Delta f(x^{(k+1)})\| \le \varepsilon$, then the solution is $x^{(k+1)}$. Otherwise, by BFGS revised formula set B_{k+1}, move on to the third step.

Among them $\Delta f(x)^{(k)}$ expresses gradient vector of the function f in x_k point.

2.3 Convergence properties analysis of BFGS algorithm

$f(x)$ are assumed to be twice continuously differentiable, by multivariate functions TAYLOR the following approximation is established:

$$\nabla f(x^{(k)}) \approx \nabla f(x^{(k+1)}) - \nabla^2 f(x^{(k+1)})(x^{(k+1)} - x^{(k)})$$

Therefore, to make B_k in approximation with $\nabla^2 f(x^{(k)})$ one kind of reasonable taking is: using B_{k+1} to replace for $\nabla^2 f(x^{(k+1)})$. B_{k+1} satisfies the equation:

$$B_{k+1} s^{(k)} = y^{(k)} \quad (1.5)$$

Among them:
$s^{(k)} = x^{(k+1)} - x^{(k)}$,
$y^{(k)} = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$.

Theorem 1.1 $f(x)$: $R^n \to R$ are assumed to be twice continuously differentiable, and observe the following iterative process $x^{(k+1)} = x^{(k)} + d^{(k)}$; k=0,1,2…; $d^{(k)}$ is the solution to the systems of linear equations

$$B_k d + \nabla f(x^{(k)}) = 0$$

Solve $\{x^{(k)}\}$ convergence in x^*. And $\nabla f(x^*) = 0$, $\nabla^2 f(x^*)$ is positive definite, then $\{x^*\}$ super linear convergence properties work as

$$\lim_{k\to\infty} \frac{\|(B_k - f(x^*))d^{(k)}\|}{\|d^{(k)}\|} = 0$$

The proof sees the reference [3]

From this, One of principles to set B_{k+1} is to make it easy to compute. By the low order revision of B_k, B_{k+1} is obtained:

$$B_{k+1} = B_k + \Delta_k \quad \Delta_k \text{ is the matrix of one rank or two rank} \quad (1.6)$$

In the equation (1.5), if rank of Δ_k is two rank symmetrical matrices, and $\Delta_k = a_k u^{(k)} u^{(k)T} + b_k v^{(k)} v^{(k)T}$ and a_k, b_k and $u^{(k)}$, $v^{(k)}$ are non-finite real number and non-finite vector respectively, By Quasi- Newton's method, the equation (1.4) will have the following equation:

$$B_k s^{(k)} + a_k (u^{(k)T} s^{(k)}) u^{(k)} + b_k (u^{(k)T} s^{(k)}) v^{(k)} = y^{(k)}$$

Solution is

$$\Delta_k = -\frac{B_k S^{(k)} S^{(k)T} B_k}{S^{(k)T} B_k S^{(k)}} + \frac{y^{(k)} y^{(k)T}}{y^{(k)T} S^{(k)}}$$

Therefore, the BFGS revised formula is established as follows:

$$B_{k+1} = \begin{cases} B_k - \dfrac{B_k S^{(k)} S^{(k)T} B_k}{S^{(k)T} B_k S^{(k)}} + \dfrac{y^{(k)} y^{(k)T}}{y^{(k)T} S^{(k)}} \\ B_k \end{cases} \quad (1.7)$$

By (1.7) we can get the following conclusion:

Suppose B_k is symmetric positive definite, B_{k+1} is determined by the BFGS revised formula; then when $y^{(k)T} s^{(k)} > 0$, B_{k+1} is symmetric positive definite.

From the deductive result, if initial matrix B_0 is symmetric positive definite, and the iteration is for any k, $y^{(k)T} s^{(k)} > 0$. If the precise linear search or the Wolfe-Powell linearity search is used in the BFGS algorithm, then in equation $y^{(k)T} s^{(k)} > 0$ is established. If the algorithm uses the AEMIJO linearity search, we cannot guarantee $y^{(k)T} s^{(k)} > 0$. In order to guarantee the symmetric positive definite of matrix B_k in the AEMIJO linear search, when $y^{(k)T} s^{(k)} \le 0$, set $B_{k+1} = B_k$.

2.4 Wolfe-Powell linear search

Make the constant σ_1, σ_2 satisfy $0<\sigma_1<1/2$, $\sigma_1<\sigma_2<1$ and take the steplength $\alpha_k>0$; Make

$$\begin{cases} f(x^{(k)} + a_k d^{(k)}) \le f(x^{(k)}) + \sigma_1 \sigma_2 \nabla f(x^{(k)})^T d^{(k)} \\ \nabla f(x^{(k)} + a_k d^{(k)})^T \ge \sigma_2 \nabla f(x^{(k)})^T d^{(k)} \end{cases} \quad (1.8)$$

equation (1.8) the condition formula of the steplength in the search.

The Wolf-Powell non-precise linear search is the expansion of AEMIJO linear search. The AEMIJO linear search is the formula (1.8) first in equation. Make initial steplength $\alpha_k = 1$, the trial steplength formula is $\alpha_k = \rho \alpha_k$. The trial steplength reduces according to the ρ proportion. If $\rho \in (0,1)$ is bigger, the change of the two neighboring trial steplength is relatively small. Many searches are needed in order to obtain the steplength α_k. If $\rho \in (0,1)$ is small, the change of the two neighboring trial steplength is relatively bigger. Fewer trial steps are needed to obtain α_k, and the steplength is possibly small. In order

to overcome this flaw, Wolfe-Powell expands the second in equation to limit the small steplength. Wolfe-Powell non-linear search trial process is as follows:

(1) If $\alpha_k = 1$ and the condition in the formula (1.8) is satisfied, the steplength $\alpha_k = 1$. Otherwise, one has to solve the aggregation $\{b\rho^i$,*i*=0,±1, ±2…. }, making the bigger establishment of the first in equation in the formula(1.8);

(2) If. α_k satisfies the second in equation in the formula (1.8), the computation is terminated and the length of step is $\alpha_k = b\rho^i$ .Otherwise, one has to solve the aggregation $\{\alpha_k^{(k)} + \rho_1^i(b_k - \alpha_k^{(k)})$,*i*=0,1,2….}, making the bigger establishment of the first in equation in the formula (2.6). Otherwise duplicate step (2).

3. BFGS parallel processing

3.1 BFGS algorithm parallel strategy

Set the initial matrix B_0 for unitary matrix, then B_0 is symmetric positive definite. The B_{k+1}, matrix produced by BFGS revised formula iteration is also symmetric positive definite. May know that the BFGS sequential algorithm has three primary tasks:(1) the solving of linear equations extracts the descending direction of the function f in x_k point, its solution is a vector; (2) From x_k and its descending direction determine the search steplength α_k; (3) Extract the new x_{k+1} point. If $\|\Delta f(x^{(k+1)})\| \le \varepsilon$ is satisfied, then the solution is x_{k+1}; otherwise calculate the BFGS revised formula and solve the matrix B_{k+1}.

In the parallel processing of the BFGS algorithm, if the parallel platform has P processors, master-slave style is used for data dispatch. We propose to solve the descending direction $d^{(k)}$ with row interleaved format parallel decomposition of CHOLESKY, Each processor is responsible for n/p component computation, the result of which will be sent to the master node. The synchronous parallel Wolfe-Powell non-linear search selects the synchronized parallel method to solve the length of step α_k. First judge whether the trial length of step satisfies the condition from the node. The lengths of step satisfying the condition will be sent back to the host node, where the lengths of step will be computed. If this maximum length of step satisfies the second formula of the formula (1.8), then the steplength is obtained; otherwise continue to circulate the solution. The BFGS revised formula parallel processing will be realized by the multiplications of the parallel matrix and the vector; the multiplications of vectors, the multiplications the vector and the matrix, and the and the matrix multiplications of matrixes.

1) Parallel CHOLESKY decomposition

In order to solve effectively the linear equations (1.4) and obtain the descending direction vector, we according to the characteristics of symmetric positive definite of the coefficient matrix in linear equations, use the parallel CHOLESKY resolution to solve the problem.

Theorem 2.2 If B is the *n* symmetric positive definite matrix, and in B all master-sub order is not a zero, then B may have the unique solution.

$$B = LUL^T \tag{2.1}$$

$$L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & 0 & \\ l_{31} & l_{32} & 1 & & \\ \cdots & \cdots & \cdots & \cdots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} u_1 & & & & \\ & u_2 & & 0 & \\ & & u_3 & & \\ & 0 & & \cdots & \\ & & & & u_n \end{pmatrix}$$

That is: L is lower triangular matrix, U is diagonal matrix. Decomposition matrix L, U element l_{ij} , u_i , for any *i*=1.2.… n will be obtained respectively by the following formula:

$$l_{ii} = 1,\ l_{ij} = (b_{ij} - \sum_{k=1}^{j-1} l_{ik} u_k l_{jk}) \quad (i=1,2,\ldots,n;\ j=1,2,\ldots,i\text{-}1) \tag{2.2}$$

$$u_i = b_{ii} - \sum_{k=1}^{i-1} l_{ik} l_{ik} u_k \quad (i=1,2,\ldots,n) \tag{2.3}$$

Set the linear equations $B_k d + \nabla f(x^{(k)}) = 0$, transform the solution system of equations $Ly = -\nabla f(x^{(k)})$ and $ULd = y$ Solution.

Solution to formula $Ly = -\nabla f(x^{(k)})$ is:

$$\begin{cases} y_1 = -\nabla f(x_1^{(k)}) \\ y_i = -\nabla f(x_i^{(k)}) - \sum_{k=1}^{i-1} l_{ik} y_k \quad (i=2,\ldots,n) \end{cases} \tag{2.4}$$

$ULd = y$ computing formula is：

$$\begin{cases} d_n = y_n / u_n \\ d_i = y_i / u_i - \sum_{k=i+1}^{n} l_{ki} d_k \ (i=n\text{-}1,n\text{-}2,\ldots,2,1) \end{cases} \tag{2.5}$$

If we want to solve L sub-triangle matrix element l_{ij} and u_i, we use row interleaved. The *myid* processor processes n the mold p complement of a number is *myid*(0≤*myid*≤*p*-1) row; the *myid* processor extracts element l_{ij} and u_i afterwards, send it to the *myid* +1 processor; the *myid* +1 processor solve the next row l_{ij} and u_i element. Then *P*

processors' average computing time is $\frac{n(n-1)}{2p}+\frac{n}{p}$.

After the solving of L and the U matrix elements, solve the equations $Ly=-\nabla f(x^{(k)})$ and $ULd=y$ respectively by using row interleaved as the formula (2.4) and (2.5) .In the formula (2.4) starting from No.0 node ask for y_1 value, then send it to No.1 node which will ask for the value of y_2, p processors will do like this in turn. In *myid*(0≤*myid*≤*p*-1) node, ask for the value of y_{myid+1}, and send it to *myid*+1 node. The formula (2.5) will get, according to the inverted order, the vector element. In *myid*(0≤*myid*≤*p*-1) node, ask for the value of d_i （*i*=*n*，*n*-1，…，2，1）element in turn, and send it to No. *myid*+1 node. After computations of each node have completed, the vector element value will be calculated and sent to the master node. The function f in x_k point descending direction vector $d^{(k)}$ will be obtained. The computing time of each node in two equations is $\frac{2n}{p}$.

2) WOLEF-POWELL non-linearity search synchronization parallel

The steplength computation has two algorithms: the precise linear search and the non-precise linear search. If the question to be solved is a single peak function or the function is quite simple, golden section or direct method of the precise linear search can be used; in the function of many variables, we used the Wolfe-Powell non- precise linear search algorithm to solve the steplength. Its basic thought is: as to the vector and its descending direction vector in some known place, suppose the length of step is $\alpha_k=1$. If（2.6）formula is satisfied, then the steplength is 1. Otherwise, from the slave node the steplength $b\rho_{myid}$ （$b>0$）， $\rho\in(0,1)$ which conforms whit the first inequality in the formula (2.6) should be sent respectively back to the host node where the maximal one will be found out. If the maximal one conforms to the second inequality in formula (2.6), the step length will be found out; otherwise the step length in every slave node which conforms to the first inequality in formula (2.6) from $\alpha_k^{(k)}+\rho_1^i(b_k-\alpha_k^{(k)})$ （*i*=1,2,…）should go back to the host node. Then the host node will judge whether they conform to the second inequality. In the process of parallel calculating, there is a process in which the host node waits for the middle results that are passed back by every minor node, namely a synchronization parallel awaiting process.

3) Parallel steps of BFGS revised formula

Proposition 1:

Provided B_k is symmetric positive definite, B_{k+1} is determined by BFGS revised formula(1.6)； when and only when $y^{(k)T}s^{(k)}>0$， symmetric positive definite is obtained.

Prove: (see reference [4]).

Proposition 2:

Provided we apply accurate linear search or Wolfe-Powell linear search in BFGS calculating, we get the inequality of $y^{(k)T}s^{(k)}>0$.

Prove: for accurate linear search $\nabla f(x^{(k+1)})^T d^{(k)}=0$,so $y^{(k)T}s^{(k)}>0$.

For Wolfe-Powell linear search : we can infer from the second inequality in (2.7)

That

$$y^{(k)T}s^{(k)}=\alpha_k(\nabla f(x^{(k+1)})-\nabla f(x^{(k)}))^T d^{(k)}>0$$

prove ended.

BFGS revised formula's role is to calculate the next symmetric positive definite matrix. Because of the application of Wolfe-Powell linear search, $y^{(k)T}s^{(k)}\succ 0$ is guaranteed. So we can apply the first inequality in (1.7) to calculate matrix B_{k+1}.

The process of the calculation of BFGS revised formula consists of three parts. The first part of formula (1.7) is B_k, and the second is $\frac{B_k s^{(k)}s^{(k)T}B_k}{s^{(k)T}B_k s^{(k)}}$. For the numerator, the calculation of each processor is that: divide n step matrix into n/p rows, and multiply them by n dimension vector $s^{(k)}$. Each processor is responsible for the multiplication of n/p rows and vector $s^{(k)}$, and gets the temporary n/p dimension vector $L^{(k)}$;multiply $L_{myid}^{(k)}$ by n/p elements transpose vector $s^{(k)T}$, and we will get another temporary *n/p* × *n/p* dimension matrix $G_{myid}^{(k)}$. Meanwhile, we divide B_k according *n/p*×*n/p* parts, each processor carries out the multiplication of the two n/p×n/p step matrix. resulting in a new *n/p* × *n/p* maxir $G1_{myid}^{(k)}$ and they are sent back to the host node to make up n step matrix. For the denominator, each processor is responsible for the multiplication of n dimension transpose vector $s^{(k)T}$ by B_k n/p column of n step matrix, and gets a temporary n/p dimension vector $L1_{myid}^{(k)}$. Divide vector $s^{(k)}$ according to n/p elements and multiply them by the temporary n/p dimension vector, plus all the results, and we get a value q_{myid}, which will be sent back to the host node by each processor. The host node adds them all and the denominator gets value q. The host node will multiply n step matrix in the numerator by $\frac{1}{q}$, then we get the result of the second part of formula (1.7). For the third part of formula (1.7) $\frac{y^{(k)}y^{(k)T}}{y^{(k)T}s^{(k)}}$, divide $y^{(k)}$ and $y^{(k)T}$ into n/p elements respectively. Then each processor multiply n/p elements by the n step transpose matrix of n/p elements, and gets a $G2_{myid}^{(k)}$, which will be sent back to the host node to make up a n step matrix .The denomination has the same division. Multiply *n/p* transpose vector elements by n/p vector element and add up all the results, and we get value w_{myid}, which will be sent back to the host node. The host node adds them all and the denominator gets *w*. The

host node will multiply $\frac{1}{w}$ by the n step matrix in the numerator, and then we get the result of the third part of formula (1.7). In the end, divide the n step matrix of the three parts into *n/p* rows respectively. Each processor is responsible for the addition and subtraction of the corresponding element in n/p rows, the results of which will be sent back to the host node to make up the next B_{k+1} to be solved in the n step matrix.

To sum up, BFGS parallel algorithm mainly includes three parts: parallel solving systems of linear equations, parallel solution of steplength, and parallel solution of BFGS revised formula. Its main calculation is as follows:

Step 1. Provided p is the number of parallel processors, accuracy ε >0,*k*=0, take primary value $x^{(0)} \in R^n$ calculation range of the host node is $\|\Delta f(x)^{(0)}\|$ .If $\|\Delta f(x)^{(0)}\| \le \varepsilon$, the calculation ends and get the solution $x^{(0)}$ of the problem. Otherwise, make the primary symmetric positive definite matrix $B_0 \in R^{n\times n}$ to be the unit matrix.

Step 2. Apply row interleaved format parallel decomposition of CHOLESKY in solving the system of linear equations to find out the descending vector $d^{(k)}$.

$$Ly = -\nabla f(x^{(k)}) \text{ and } ULd = y$$

Step 3. Use the synchronous parallel Wolfe-Powell non-linear search to set the steplength α_k.

Step 4. Provided *k*=*k*+1, in each node do the parallel calculation of n/p elements $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ and then send them back to the host node which will calculate $\|\Delta f(x^{(k+1)})\|$. If $\|\Delta f(x^{(k+1)})\| \le \varepsilon$, the solution is $x^{(k+1)}$;otherwise, calculate B_{k+1} by using parallel calculating BFGS revised formula, and turn to 2.

3.3 The BFGS parallel algorithm analysis

The BFGS parallel algorithm analysis includes the analysis of the complexity of calculation and communication. Because the analysis of the complexity of communication relies on the systematic structure of the computer, this paper mainly carries out the analysis of the complexity of calculation. Provided B is n step matrix and there are P processors (nodes), the task for each processor is n/p rows. Considering that B is n step Symmetric positive definite matrix, if divided by part, the task for the processors will not be in balance. So we apply row interleaved division mode to divide the task for processors, which will enable the load of each processor to be relatively in balance. Calculating time complexity mainly consists of three parts:

(1) The time for calculating systems of linear equations:

Sequential calculation time: $T_1 = \frac{(n+1)(2n+1)}{6}$

Parallel calculation time:

$$T_{p1} = \frac{n(n-1)}{2p} + \frac{n}{p} + \frac{2n}{p} = \frac{n(2n+1)}{2p}$$

(2) The time for calculating steplength:

Sequential calculation time: $T_2 = 2n(4n-1)$

Parallel calculation time: $T_{p2} = \frac{4n-1}{p}$

(3) The time for calculating BFGS revised formula:

Sequential calculation time:

$$T_3 = 2n(n^2 + 4n - 1)$$

Parallel calculation time:

$$T_{p3} = \frac{n(n^2+4(p^2+p)n-2p^2)}{p^3}$$

Total time for sequential calculation :

$$T_s \approx 2n^3 + 16n^2 - 3n$$

Total time for parallel calculation:

$$T_p \approx \frac{n^3+6p^2n^2+5p^2n}{p^3}$$

Parallel speedup rate :

$$S_p = \frac{T_s}{T_p} = \frac{p^3(2n^3+16n^2-3n)}{(n^2+6p^2n+5p^2)} \to 2p^3 (n \to \infty)$$

From the formulas above we can see that if we don't consider the cost of communication, and the variable of function f is big enough, BFGS parallel algorithm will have a ideal accelerating rate.

4. NUMERICAL VALUE TEST

The following test results from four PCs of the same structure. N stands for the scale of function variable dimension, and P the number of processors. Applying MPI and C language program design, we get the result of BFGS parallel algorithm. Its test function is as follows[5]:

Problem 1:

$$f(x) = \sum_{i=1}^{n/2} \{100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2\}$$

Table 1. the Numerical Results for Problem one

n \ p	Run times（second）		
	1	2	4
400	52.74649	3.347211	2.239653
1000	187.42072	11.54893	3.407536
2000	1515.94801	94.759147	12.534102

Problem 2: $f(x) = (\sum_{i=1}^{n} i x_i^2)^2$

Table 2. the Numerical Results for Problem two

n \ p	Run times（second）		
	1	2	4
400	63.315709	5.813523	2.144065
1000	553.428127	33.213906	5.125713
2000	1624.23901	101.037254	13.041233

From the tables above, we can see that with the increase of parallel processors node, the working time of the system will decrease and the accelerating rate will be relatively higher.

5. CONCLUSION

BFGS algorithm can use unit matrix as the primary symmetric positive definite matrix. Each step in the steplength nonlinear search can be tested by using $\alpha_k = 1$ as the primary steplength. On the other hand, we can make full use of the characteristic of matrix symmetric positive definite to reduce the use of RAM space so as to increase the utilization ratio of RAM space. Therefore, BFGS algorithm is very suitable for the parallel processing of computers. The test result shows that BFGS parallel algorithm increases the speed of solving the problem of large-scale unconstrained nonlinear optimization with a ideal parallel speedup rate.

6. REFERENCES

[1] Y.Dai,"Convergence properties of the BFGS algorithm". SIAM Journal on Optimization.13(2003).697~701

[2] D.H.Li, M.Fukushima, "A modified BFGS method and its global convergence in nonconvex minimization.Journal of Computational and Applied Mathematics", 129(2001).15~35

[3] D.H.Li, X.J.Tong, et al, "Number Optimization (M). BeiJing"：Science Publishing House，2005

[4] D.H.Li, M.Fukushima, "On the global convergence of BFGS method for nonconver unconstrained optimization problems". SIAM Journal on Optimization. 11(2001). 1054~1064

[5] F.Y.Zheng, "Research on some parallel algorithms for optimization problems". The degree of Master of Science from Shandong University of Science and Technology.2004

Ahmutex: An Ad-Hoc Distributed Mutual Exclusion Algorithm Based On Read/Write Character

Zheng Wang, Xinsong Liu, Meian Li
8010 R&D Univ. of Elec. S&T of China 610054
Chengdu, Sichuan, China
Email: wangzheng151400@163.com

ABSTRACT

Distributed mutual exclusion is an important problem for Ad hoc networks. Based on the analysis of traditional distributed mutual exclusion algorithms, a novel distributed mutual exclusion algorithm called AHMUTEX based on read/write character was proposed for Ad hoc networks. The algorithm, which was constructed with read/write character and based on the token-asking algorithm, utilized the Lamport's logical timestamp method to ensure the time sequence and the validity of control messages. Furthermore the Lamport's timestamp was employed to prevent nodes from starvation. Then AHMUTEX reduced the response delay of Ad hoc networks through the methods of message multiplexing and distinguishing read/write operation sets. Also it made a trade-off between the message complexity and the response delay. Performance analysis and simulation results show that the algorithm can be accommodated to the requirements of Ad hoc networks. Compared with traditional algorithms, it has lower message complexity, shorter response delay and better fairness.

Keywords: Author Guide, Manuscript, Camera Ready Format, and Instructions for Authors, Paper Specifications.

1. INTRODUCTION

An Ad hoc network is a collection of nodes that communicate over paths composed of one or a sequence of wired/wireless links. In an Ad hoc network, the node mobility pattern creates unpredictable links formation and removal [1,2]. As a consequence, a path between two nodes can change very frequently due to topology change. The distributed mutual exclusion is described as that only a single process or node can be allowed to a shared resource termed as a critical section or CS. Critical section can be a portion of codes or some physical equipment and so on. In an ad hoc network, all nodes sometime need to employ some critical sections concurrently. For an example, America DARPA（Defense Advanced Research Projects Agency） recently stakes an Ad hoc network project presented as the self-resume landmine field system. In future, the system can employ intelligent landmines to construct the defense line for tanks. When a tank enters into the landmine field, all landmines in the field must select a representative to deal with it. And several landmines will be wasted without distributed mutual exclusion algorithm for ad hoc networks.
In this paper, we present a novel distributed mutual exclusion algorithm called AHMUTEX (Ad Hoc MUTual EXclusion) for ad hoc networks based on read/write character. The remainder of the paper is organized as follows. Section 2 makes a summary for traditional algorithms and gives a distributed mutual exclusion model for ad hoc networks. In section 3, a concept of read/write character is given and a novel distributed mutual exclusion algorithm is proposed. In section 4, correctness and fairness of the algorithm are proved. The section also makes an analysis for the message complexity and response delay. And simulation results will be given in section 5. Finally, Section 6 concludes the paper.

2. PRELIMINARY

In the section, we review traditional exclusion mutual algorithms. Furthermore, we gave these disadvantages preventing these algorithms from adapting themselves to Ad hoc environments. At last a novel model for ad hoc mutual exclusion was given.

Traditional mutual exclusion algorithms could be either token-based or nontoken-based [3, 4, 5]. Usually they had to make a trade-off among message complexity, response delay and so on. Lamport's distributed mutual exclusion algorithm was an example of nontoken-based mutual exclusion algorithms [6]. And it was widely employed in Ad hoc systems. The message complexity of it was 3*(N-1) and the synchronization delay was Ricart&Agrawal (RA) proposed two types of distributed mutual exclusion algorithms. All of them achieve the O(N) message complexity [5]. Over the last decade, quorum-based mutual exclusion algorithms, which were a generalization of Maekawa's algorithm, had been paid considerable attention. Maekawa cleverly constructed quorums that could decrease the message complexity to $O(log^N)$. But few mutual exclusion algorithms were researched for them. In token-based mutual exclusion algorithms for Ad hoc, a unique token existed in the network and only the holder of it could enter the CS. Examples of mutual exclusion algorithms for Ad hoc were J. Walter's and A. Virgillito's algorithms and so on [6,7]. Like Lamport's, nontoken-based mutual exclusion algorithms exchanged messages to determine which nodes could enter the CS in an Ad Hoc network. A representative of them was J.R. Jiang's [2]. Though so many distributed mutual exclusion algorithms had been researched, few of them could adapt well to Ad hoc networks in arbitrary topologies. The algorithms required nodes ceaselessly to apperceive all other nodes, which made Ad hoc nodes waste their energy and limited communication bandwidth.

According to the analysis results, we gave a distributed system model for Ad hoc networks. Our system model also consisted of N processes termed as $\{P_i \mid 0 \leqslant i \leqslant N\text{-}1\}$. Without loss of generality, there were no global memory and no global clock in the system mode [8,9]. Nodes were not directly connected, but they could communicate with each other over mutli-hop paths. The network diameter was K and it could be utilized to calculate the message complexity, space complexity and response complexity of algorithms. The underlying communication medium was reliable and

sites did not crash. The system model also assumed that the message underlying channels were FIFO. Then T was the average message delay between two nodes without intermediate nodes. On the other hand, E was the average delay of a CS execution. Specifically, the content of a message could not be changed, when it was delivered among several nodes. While a process was waiting to enter its CS, it could not send another request for the CS. So the distinct requests in an Ad hoc network could be termed as $\{R_i \mid 0 \leqslant i \leqslant \text{N-1}\}$. In addition, $Pr(R_i)$ was the priority of the request R_i. According to these notations above, several definitions could be given.

Definition1. Concurrent: R_i and R_j were concurrent iff R_i was received by P_j had sent R_j and R_j was received by P_i after P_i had sent R_i.

Definition2. ConcurrentSet (Cset):$\text{Cset}_i = \{R_j \mid R_j$ was concurrent with $R_j\} \cup \{R_i\}$.

Definition3.ConcurrentHeader(Cheader): _.

$R_i : \Pr(R_i) > \Pr(R_j) \quad \forall R_i, R_j \in Cset_k \quad i \neq j$;

Definition 4. ConcurrentTailer (CTailer):

$R_i : \Pr(R_i) < \Pr(R_j) \quad \forall R_i, R_j \in Cset_k \quad i \neq j$;

Definition 5. Pred:

$$\Pr ed(R_i, Cset_k) = R_j \quad iff \quad R_j \in Cset_k \ \wedge \ \Pr(R_i) < \Pr(R_j) \ \wedge \ \neg(\exists R_k \in S \mid (\Pr(R_i) < \Pr(R_k) < \Pr(R_j)))$$

Definition 6. Succ:

$$Succ(R_i, Cset_k) = R_j \quad iff \quad R_j \in Cset_k \ \wedge \ \Pr(R_i) > \Pr(R_j) \ \wedge \ \neg(\exists R_k \in S \mid (\Pr(R_i) > \Pr(R_k) > \Pr(R_j)))$$

Definition 7.

$$GV(R_i, R_j) = \{R_k \mid \Pr(R_i) \geq \Pr(R_k) \geq \Pr(R_j)\}$$

Definition 8.

$$Dist(R_i, R_j) = |GV(R_i, R_k)| - 1$$

Definition 9. Hop(P_i,P_j): In an Ad hoc network, it was equivalent to the number in which nodes switch a message transmitted from P_i to P_j. The definition could be used to evaluate the message complexity of AHMUTEX and the traditional. The definition 9 is unsymmetrical and *Hop(P_i,P_j)* was not generally equivalent to *Hop(P_j,P_i)*.

Definition 10. RCset: $RCset_i = \{R_j \mid R_j$ was concurrent with $R_j\} \cup \{R_i\}$ and for all $R_k \in RCset_i$ R_k was with read character.

3. AHMUTEX ALGORITHM

3.1 Issued idea

The algorithm assumes the same system model as the one presented above. Also it supposes that an underlying Ad hoc network channel is FIFO. P_i saves a request queue containing request messages in the order of priorities, received by it. In our algorithm, all requests are totally ordered by the priority and queued by all processes in the order. As following as, we research the algorithm in several cases. Initially, there has been a token owned by a process in the Ad hoc networks. And the procedure of selecting a process to own the token has been discussed in many researches.

Light Load Case: In the case, all request queues are null and the process requesting the CS broadcasts a request message in the Ad hoc network, similar the traditional the token-asking algorithm. A process receiving a request message can immediately determine whether the requesting process should be allowed to enter the CS and put the request message into the RQ_i queue proposed above. When a process owning the token doesn't request the CS, the process can send the token message to the requesting process. When a process requesting the CS receives the token message, it enters the CS.

Heavy Load Case: In the case, several nodes requesting the CS broadcast respective request messages and invoke mutual exclusion in an Ad hoc network. In all nodes, the RQ_i queues corresponding to these messages are ordered by the timestamps that are carried by the request messages. When a node owning the token is out of the CS, it can send the token message to the R_i whose priority is the highest in the messages sent to the node. The P_i receiving the token message, it set the local variable *Has_Token* to True. According to as above, the requests with read character can synchronously enter the CS, which can reduce the synchronous delays. Here, a definition of read concurrent sets is given as above.

That is to say, $Rcset_i$ is a subset of $Cset_i$. In our algorithm, RQ_i is consisted of several *RCset*s and write operation requests. When Pi executing read operation in the CS receives several requests with read character, RCseti is constructed according to the definition 9 and ordered by the Lamport's timestamp. Note that the $RCset_i$ is being constructed until a request comes with write character. Pi sends *virtual_token*s to the nodes with read character in the $RCset_i$ and P_k receiving the *virtual_token* message enters the CS for read operation. When P_k leaves from the CS without the token, it sends a release message to the *CHeader* of $RCset_i$. After all processes in $RCset_i$ leaves from the CS, the token may be sent to either the *CTrailer* of $RCset_i$ or the *Cheader* of the next *Cset*. In the whole procedure for mutual exclusion, all requests message received by Pi are inserted into RQ_i ordered by Lamport's timestamps. In the case, an AHMUTEX example in an Ad hoc network has been illustrated in Fig. 1.

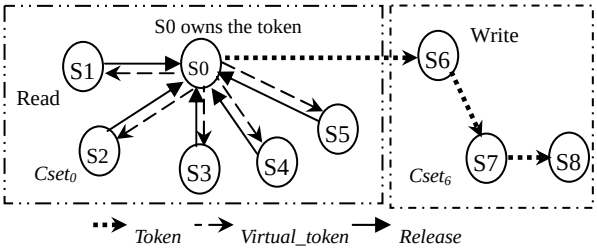


Fig.1 An example of AHMUTEX algorithm

3.2 Control messages and data structure

In Ad hoc systems, short frame messages must be employed since the bandwidth of Ad hoc networks is limited compared with others. Similar to them in the traditional, the message formats, which are in our algorithm, are as follows. And the general message format is defined as: *msg(sender, receiver, parameter);*

request(i,all,timstamp,rw): a *request* is sent by the process P_i for the CS with the timestamp, which includes the form of

(sn, i). The *request* messages are broadcasted to all processes in an Ad hoc network. Given all other processes receive the *request* message. And *rw* is define as *read/write* character indicating that the process wants either read operation or write operation. And a process can't leave the CS before it completes write/read operation requested.

token(i,j): a *token* message indicates that P_i grants P_j's *request* to enter the CS and to execute read/write operation and P_j becomes the holder of *token*.

vtoken(i,j): a *vtoken* message indicates that P_i grants P_j's *request* to enter the CS and to execute read operation. And a node doesn't set the *Has_Token* to True, when it receives a *vtoken* message.

release(i, j): when a process included in a read sequence $Cset_i$ leaves the CS, it will send a *release* message to the R_j of the $Cset_i$. And the message comes from Quorum-based algorithms presented by.

A process P_i maintains the following local data structures:

A local Boolean variable: $RW\_Charecter_i$. And it indicates that process P_i wants to execute read operation or write operation.

A local integer variable: $Local\_Sequence_i$. And it is employed to implement the Lamport's timestamp. According to the Lamport's algorithm presented above, $Local\_Sequence_i$ is equivalent to the local counter C_i. As same as $Local\_Sequence_i$, a local variable $Max\_Sequence_i$ is equivalent to the T_i.

A local variable: Has_Token_i. And it can be a Boolean variable. When it is True, the process P_i owns the token of the Ad hoc network for the CS.

A local variable: $Vtoken\_counter_i$. And it records the number of $|Rcset_i|$-1 in the process P_i. Furthermore it is increased by the read *request* messages and decreased by the *release* messages. Specifically, a local variable $Tsender_i$ records the sender of *vtoken* and *token* messages.

A request message queue: RQ_i. The queue saves all *request* messages that are ordered by the Lamport's timestamp and the identifiers of R_i. Process P_i, which owns the genuine token, can achieve the $Cset_i$ from the queue.

3.3 Algorithm description

The novel algorithm was outlined on follow. Procedures were droved by messages/events. And each procedure was executed atomically when a message was received or sent. Without loss of generality, the execution of requests for the CS in AHMUTEX was always in the order of their decreasing priority. For each CS access, there were exactly a request message broadcasted and a token message for permission.

Initial local variables for process P_i:

- integer $Local\_Sequence_i = 0$ and $Max\_Sequence_i = 0$ and Tsenderi = NULL
- boolean Has_Token_i = False
- boolean $RW\_Charecter_i$ = False
- integer $Vtoken\_counter_i = 0$
- queue of request messages RQ_i = NULL

OnInvMutEx: Process P_i requesting for the CS executes the following to invoke mutual exclusion:

- $Local\_Sequence_i = Max\_Sequence_i + 1$
- If P_i wants to read then $RW\_Charecter_i$ = False else $RW\_Charecter_i$ = True
- $request_i$ =*(i, all, $Local\_Sequence_i$, $RW\_Charecter_i$)*, where "*all*" is broadcast address
- Broadcast requesti to all processes including itself

OnRcvReq: Process P_i receives a request message from P_j, where R_j = *(j, all, $Local\_Sequence_j$, $RW\_Character_j$)*

- $Max\_Sequence_i$ = $MAX(Max\_Sequence_i, Local\_Sequence_j)$
- Insert R_j into RQ_i in the sorted order by the Lamport's timestamps
- If not *Has_Token* then exit from the procedure
- If *Has_Token* and ($RW\_Character_i$ or $RW\_Character_j$) and (P_i in the CS) then exit from the procedure
- If *Has_Token* and not($RW\_Character_i$ or $RW\_Character_j$) and (P_i in the CS) then send *Virtual_token* to P_j and $Vtoken\_counter_i$ = $Vtoken\_counter_i + 1$
- If *Has_Token* and not (P_i in the CS) send token message to P_j

OnRcvRle: Process P_i receives a release message from P_j, where *release=(i, j)*

- $Vtoken\_counter_i = Vtoken\_counter_i - 1$
- Delete the element Rj of RQi
- If *Has_Token* and not(RQ_i is NULL) and ($Vtoken\_counter_i$ == 0) then send *token* message to the $CHeader_i$ of $CSet_i$ and *Has_Token* = False

OnRcvToken: Process P_i receives a token message from P_j

- *Has_Token* = True and $Tsender_i = P_j$
- Delete some elements of RQ_i, where Pr(element)>Pr(R_i)
- If (R_i in RQ_i) then P_i enter the CS
- If $|RCset_i| > 1$ then send *Virtual_token* to the processes of $RCset_i$ and $Vtoken\_counter_i = |RCset_i|-1$

OnRcvVToken: P_i receives a *Vtoken* message from P_j

- Delete some elements of RQ_i, where Pr(element)>Pr(R_i)
- P_i enter the CS and $Tsender_i = P_j$

OnExCS: Process P_i leaves from the CS

- If not Has_token then send *release(i,j)* message to $Tsender_i$
- Delete the element R_i of RQ_i
- If *Has_token* and not(RQ_i is NULL) and ($Vtoken\_counter_i$ == 0) then send token message to the $CHeader_i$ of $CSet_i$ and *Has_Token* = False

4. PROOFS AND PERFORMANCE ANALYSIS

In the section, we showed that the AHMUTEX algorithm achieved mutual exclusion and it was free from starvation. Furthermore, the performance analysis was given for AHMUTEX in the section.

4.1 Correctness and Fairness

Theorem 1. Mutual exclusion in AHMUTEX can be achieved.

Proof. Assume to the contrary, two processes P_i and P_j were executing the CS simultaneously. From AHMUTEX description above, we knowed P_i's request R_i had the same priority of R_j under the case. That was to say the system state satisfied the following invariants in AHMUTEX:

$$\exists i \exists j((i \neq j) \wedge (Has\_token_i = True) \wedge (Has\_token_j = True))$$

and

$$pre(S_i,Cset_i)=Pre(S_j,Cset_i)=Succ(S_i,Cset_i)=Succ(S_j,Cset_i).$$

It was easy to see that the invariant contradicted AHMUTEX algorithm since a process would set its *Had_token* to False with sending the token to only another process. And only one process could achieve the

token. So it was impossible to achieve the first invariant above. Lamport's timestamps guarantee that $Pr(R_i)$ was not equivalent to $Pr(R_j)$ for $i \neq j$. So $Pre(S_i,Cset_i)= Pre(S_j,Cset_i)=Succ(S_i,Cset_i)=Succ(S_j,Cset_i)$ was not achieved. Thus AHMUTEX could achieve mutual exclusion.

Theorem 2. Starvation was impossible in AHMUTEX algorithm.

Proof. Starvation occured when a process waits indefinitely to enter the CS while other nodes were constantly entering the CS. The system state could be described as the following expression:

$$\neg\exists x\exists i\forall j((i \neq j) \mathrm{I}\ (x < \infty) \wedge (\Pr(R_i) + x > \Pr(R_j))).$$

Given there be a starving $\boldsymbol{P_i}$ broadcasting request messages in an Ad hoc network. Then all other nodes could receive the request since communication channels were reliable. And in the same $Cset_i$, each $Dist(P_i,P_j)$ was limited and decided by Lamport's timestamps. Furthermore in the $Cset_i$, $Pr(R_i)$-$Pr(R_j) = Dist(P_i,P_j)$. In AHMUTEX algorithm, $\boldsymbol{R_i}$ was assigned a sequence LC_i larger than all other sequence numbers according to Lamport's timestamp algorithm. And all LC_j ($i \neq j$) could be larger than LC_i in AHMUTEX algorithm after receiving R_i. R_j was equivalent to $\boldsymbol{LC_j+1}$. $\exists x\exists i\forall j((i \neq j) \wedge (x < \infty) \wedge (\Pr(R_i) + x > \Pr(R_j)))$ contradicts the variant above. Thus $\boldsymbol{P_i}$ entered the CS after anther when $Dist(P_i,P_j)$. Starvation was impossible in AHMUTEX algorithm.

4.2 Performance analysis

The performance of mutual exclusion algorithms was generally measured by message complexity (average message number) and response delay. To simplify the problems, we only take account of influence of the token-asking machine instead of the read/write characters.

1) Message complexity: In our algorithm, P_i broadcasted a request message for a CS execution. And the message over the Ad hoc network traverses a tree or a connected graph. We did not discuss the methods of traversing an Ad hoc network. So we gave an assumed function as $BroadcastCost(P_i)$ for the cost of traversing an Ad hoc from a node P_i. Then the *token* holder P_j received R_i and sent the token to P_i. Here, P_j had knowledge of P_i and communicated with it in Point-to-Point model, where the *token* message number transmitted and switch was $Hop(P_j,P_i)$. And the intermediate node served as routers for P_i and P_j. So in AHMUTEX, average message number for a CS execution was $BroadcastCost(P_i)+ Hop(P_j,P_i)$. Compared with our algorithm, Lamport's needed $BroadcastCost(P_i)$ request messages and $\sum_{j=1;i\neq j}^{N} Hop(P_j,P_i)$ reply messages for a CS execution. And $Hop(P_j,P_i) < \sum_{j=1;i\neq j}^{N} Hop(P_j,P_i)$. If Lamport algorithm needed the average numbers of $\sum_{j=1;i\neq j}^{N} Hop(P_j,P_i) + \sum_{j=1;i\neq j}^{N} Hop(P_i,P_j)$ without broadcast.

2) Space complexity: Traditional algorithms generally needed the fixed size *request* queues to storage *request* messages. And the sizes of *request*/*reply* queues were equivalent to the numbers of processes in the system models. In the traditional system model, the size of *request*/*reply* queues was N. In AHMUTEX, P_i would delete the all elements R_j of the RQ_i, when R_j occurred and $Pr(R_i)<Pr(R_j)$. In the worst case, a process P_i didn't request the CS for a long time. And all other processes had requested the CS. Then P_i received all other R_j and the RQ_i had the size of N-1. In the best case, P_i sent *request* messages and the size of RQ_i was zero. Due to many uncertain reasons such as network diameters, space complexity was difficultly evaluated in AD hoc networks. But it was obvious that space complexity was lower in AHMUTEX than in the traditional.

3) Response delay: The response delay was the time required after a node broadcasted its requests and before it exited the CS. In the section, we discussed the response delay in two cases:

Light load case*:* P_i requested the CS and the request message took the time cost of $Hop(P_i,P_j)\times T$, where P_j was the token holder. On the other hand, P_j sent the token to P_i and the token message took the time cost of $Hop(P_j,P_i)\times T$. The response delay could be described as following : $(Hop(P_i,P_j)+Hop(P_j,P_i))\times T+E$. In Lamport's algorithm, P_i needed to wait until all other nodes received *request*s and all others gave permission to it. In our system model, the network diameter was K. So the response delay could be described as following: $2\times K\times T+E$. It was obvious that K was larger than all $Hop(P_i,P_j)$. That was to say the response delay of AHMUTEX was relation to the hop number of the requesting node and the token holder, which made it perform better than the Lamport's.

Heavy load case*:* Several processes invoked mutual exclusion in an Ad hoc network. It was difficult to evaluate the response delay since many messages may parallel the transmission time of them in the network. Here, we described the global time of a $Cset_i$ to evaluate the response time. And the time cost could be given by the following expression:

$$\begin{aligned}&|GV(Cheader_k,Ctailer_k)|\times E+\sum_{R_i\in Cset_i \wedge R_i\in Cheader_i} Hop(\Pr e(R_i),R_i)\times T\\&+(Hop(Cheader_k,the\ \ holder\ \ of\ \ token)\\&+Hop(the\ \ holder\ \ of\ \ token,Cheader_k))\times T\end{aligned}$$

Nodes had to wait for replies from all others in Lamports's algorithm. So no response delays overlapped in the mutual exclusion procedure. And the response delay of a $Cset_i$ could be : $(2\times K\times T+E)\times|GV(Cheader_k,Ctailer_k)|$. It was obvious that the response delay was larger in Lamport's than in AHMUTEX.

5. SIMULATION STUDY

The simulation of the algorithm in the paper had been carried out using the NS2 simulation environment. In the section, we gave the simulation settings and results. The simulation environment was composed by several nodes randomly spread over an area of 600×600 meters. The transmission radius of each node was 200 meters and the diameter of the Ad hoc network was 3. Each hop delay and each CS execution were equivalent to T. And each node issues request message randomly, using Possion distribution. Furthermore, other simulation settings conformed to the system model presented above.

Fig. 2.(a) shows the number of messages transmitted in an Ad Hoc network having fixed tree topology, but the number of nodes change between 10 and 20. In order to simplify

simulation operations, the simulation results are achieved in the light load case presented above. Fig. 2.(b) shows the average numbers of the messages transmitted in an Ad Hoc network, in which the network topology is arbitrarily organized and the number of nodes is 20. Furthermore, each message transmitted in the network can be summed up for the message number. The results of the simulation confirm that our algorithm performs better than Jiang's. As presented above, our algorithm makes a take-off between message complexity and synchronization delay by read/write character. When the sizes of RCSet are greater than 1, end2end communications are only needed to send release messages. One the other hand, Jiang's needs (N-1) requests and (N-1) replies for per CS. Though in Ad hoc networks, we can broadcast requests for Jiangs's, but it's reply messages have to be transmitted in end2end mode. So in multi-hop networks such as Ad hoc networks, Jiang's has higher message complexity than the novel does so.

Comparing Fig. 3.(a) with Fig. 3.(b), the results show that dynamic topologies have greater influence on Jiang's than on AHMUTEX. And all message costs in the simulation is up to the theoretical values since some messages are repeated to transmitted in the Ad hoc for specific topologies such as tree. What is more, the average number of messages in arbitrary topologies than it in a fixed topology for Ad hoc networks.

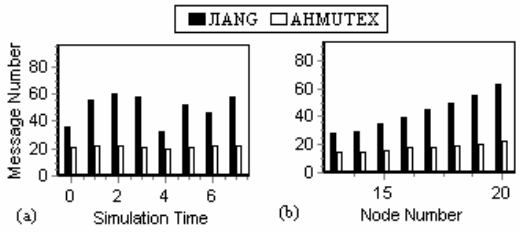


Fig. 2. Comparison of message complexities

Fig 3 shows the average delays of AHMUTEX and Jiang's algorithm in Ad hoc networks. Also the Ad hoc network has different topologies in each simulation result. In order to simplify the simulation results, we assume the CS execution time E should be equivalent to the T presented above. Similar to the simulation above, we execute it under two cases including arbitrary topologies and fixed tree topology. Processes issue read/write requests randomly, employing a Possion distribution we take one hundred pairs of delays to evaluate the average delays.

It is clear that AHMUTEX compared with Jiang's has effective performance improvement with respect to the time cost. Because of read/write character, AHMUTEX can greatly reduce the delays of mutual exclusion through parallelizing read operations. So AHMUTEX greatly suits to the mutual exclusion type including concurrent read operations in Ad hoc networks.

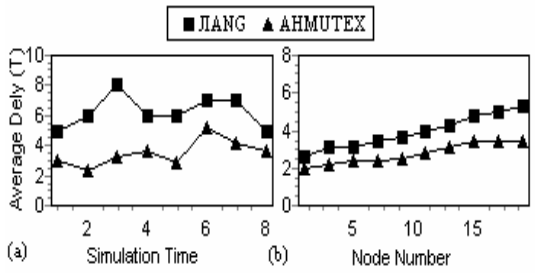


Fig. 3. Comparisons average response delays

6. CONCLUSIONS AND FUTURE WORKS

Aiming at the comparatively laggard level of the distributed mutual exclusion algorithms for Ad hoc networks, a token-asking algorithm AHMUTEX had been presented with read/write character. The algorithm combined broadcast, Lamport's timestamps and read/write character into the token-asking algorithm to improve the performance of mutual exclusion algorithms. Furthermore, it employed broadcast to deal with the dynamic topologies of Ad hoc networks. And it utilized read/write character to parallelize concurrent read operations instead of serializing. To sum up, AHMUTEX has low message complexity and short synchronization delay and response time. In future, we will discuss how to reduce average hop numbers between the holder and the requester of tokens. What is more, the mobility of nodes in Ad hoc networks should be considered as an important factor.

7. REFERENCE

[1] Y.Chen, L.W. Jennifer, Self-stabilizing dynamic mutual exclusion for mobile ad hoc networks. Journal of Parallel and Distributed Computing, 2005，65(9):1072-1089

[2] J.R.Jiang, A Distributed h-out of-k Mutual Exclusion Algorithm for Ad Hoc Mobile Networks. Proceeding of the international Parallel and Distributed Processing Symposium (IPDPS'02), 2002.

[3] L. Lamport, Time, Clocks and the Ordering of Events in Distributed Systems. Communication of the ACM, 1978,21(7): 558-565.

[4] M. Maekawa, A logN Algorithm for Mutual Exclusion in Decentralized Systems. ACM Transactions on Computer Systems, 1985, 3(2): 145-159.

[5] G. Ricart, A.K. Agrawala, An Optimal Algorithm for Mutual Exclusion in Computer Networks. Communication of the ACM, 1981, 24(1): 9-17.

[6] J. Walter, G.Cao, et al, A k-mutual exclusion algorithm for ad hoc wireless networks, Proceedings of the first annual Workshop on Principles of Mobile Computing (POMC 2001), 2001.

[7] R. Baldoni, A. Virgillito, et al, A Distributed Mutual Exclusion Algorithm for Mobile Ad-Hoc Networks [C]. Proceeding of the Seventh International Symposium on Computers and Communications (ISCC'02), 2002.

[8] J.W.Yin, P.Zhou, et al, Distributed Operating System, Changsha: National Defence Technology publishing company, 2000.

[9] I. Suzuki, T. Kasami, A Distributed Mutual Exclusion Algorithm, ACM Transactions on Computer Systems, 1985, 3(4): 344-349

The Research And Application Of The Parallel Algorithm For Qr Decomposition Of Matrix

Aimin Yang1, Chunfeng Liu1,2, Xinghua Ma1, Nan Ji1,Min An1
1.College of Science, Hebei Polytechnic University, Tangshan, Hebei Province, 063000 China
2. Hebei University of Technology, Tianjin, 300130 China
Email: aimin@heut.edu.cn, liucf403@163.com

ABSTRACT

With the rapid development of high-speed network technology, the cluster systems have been the main platform of parallel algorithm. Because of their high communication delay, some parallel algorithms of fine grain are not fit for running in this environment. Therefore, it is necessary to study their parallel achievements in cluster systems. In terms of that, a new way for QR decomposition of matrix is proposed and the coarse grain parallel algorithm is obtained. In the process of designing these parallel methods, the separately principle was based on, the original matrixes were divided into some blocks, then each block was distributed into various node machines, which run the submission in dependently. It was a much better proposal to the cluster system, which had no more nodes. At last, a simulation was given. The solution obtained shows that the designing parallel algorithm has much higher speedup in the cluster system.

Keywords: Cluster System; QR Decomposition; Parallel Algorithm; Speedup; efficiency.

1. INTRODUCTION

The luster is a high efficient parallel processing system, it is composed of a set of high performance workstations and high level pls with the supporting software, these parts of the system coordinate with each other, the rapid network makes the delay of the communication decrease greatly, but in fact, we can't overlook the delay, it affects the efficient of the parallel computing, so the algorithm under the cluster is only suitable for the middle granularity or that above it. If so, there is need to design the coarse granularity parallel algorithm, which is suitable for the network parallel computing. From this, based on the method of measuring the spending of the communication (please confer literatures [1, 2]), we still study the parallel realization of the matrix's OR decomposition under the MPI environment (please confer literature [3]).

2. QR DECOMPOSITION OF A MATRIX

QR decomposition of a matrix is to decompose the non-odding coefficient matrix to the product of an orthotropic matrix and an upper triangle matrix R (please confer literatures [4,5]):

$A = QR$

Then we can change the problem of solving common equation sets to the solution of the triangle equation sets, it can also be used to solve the total characteristic root of any real non-odding matrix.

The QR decomposition can be realized through plane revolving transform, if A is a matrix of $n \times n$, the element of the matrix can be expressed by a_{ij} $(1 \le i, j \le n)$. A standard plane revolving transform is to properly choose Q^T as the product of the limited plane revolving transform sequence, one of the transform would expunge an element under the main diagond without destroying the former zero elements, then we obtain $Q^T A = R$. In order to expunge the (p,q) elements, then we can only change the $p-1, p$ rows of the matrix A. As follows

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{p-1,1} & a_{p-1,2} & \Lambda & a_{p-1,n} \\ a_{p,1} & a_{p,2} & \Lambda & a_{p,n} \end{bmatrix}$$

Here

$$s = \frac{a_{p,q}}{\sqrt{a_{p,q}^2 + a_{p-1,q}^2}}$$

$$c = \frac{a_{p-1,q}}{\sqrt{a_{p,q}^2 + a_{p-1,q}^2}}$$

We mark the changing matrix as $Q_{p,q}$, the action of this matrix is to act on the $p-1$, p , rows of the matrix A to expunge the (p,q) element.

If $A_0 = A$, the serial computing of the QR decomposition based on the plane revolving changing can be processed as follows if according to the line sequence:

For the computation of $l = 1,2,\Lambda\ ,n-1$:

$$Q_l = \prod_{i=0}^{n-l-1} Q_{n-i,l}\ , \qquad A_l = Q_l A_{l-1}$$

Obviously,

after $(n-1)+(n-2)+\Lambda\ +1 = n(n-1)/2$ steps of change, then

$R = A_{n-1}$, $Q^T = Q_{n-1}\Lambda\ Q_1$

It is the plane revolving changing of the matrix A.

We can see that in order to expunge the (p,q) element during the plane revolving changing, we don't necessarily to act on the $p-1$、p rows. We can act on any k and p rows $(k < p)$, if only the (k, j) $(1 \le j < q)$ element is 0, the (k,q) element is not 0.

Though there are strong limits during the zero expurgation among every element, if using the plane revolving changing, we can parallelly expunge zero elements among several elements. Literature [5] gives us the parallel algorithm of the QR decomposition, which are the standard

algorithm and the gripple algorithm, the principle of them is consistent, the only differentiation is in sequencing the $(p_k, q_k), (1 \le q_k \le p_k \le n)$. For this kind of parallel, $Q_{p,q}Q_1$ only acts on two rows of the Q_L, other elements remains the same,(corresponding with the two rows of A_l in $Q_{p,q}A_l$,which participate in the zero expunction),so each time the main processor will pass two rows data of the current A and Q_l at one time (four rows date totally) to each processor, so the total time of the transmission is $n(n-1)\times 4n$,obviously, the data transmission quantum of the algorithm is not suitable for the network parallel cluster environment that with big delay communication.

3. THE PARALLEL OF THE QR DECOMPOSITION UNDER CLUSTER SYSTEM

The parallel algorithm under network cluster environment asks for less spending in the communication so, according to the parallel nature of the QR decomposition, we reparation the tasks.

We block the former matrix A: in order to get k equal parts, we decompose the elements under the diagonal into triangle and trapezium parts, (the diagonal elements are contained in the upper triangle). Then we can see the zero expunction of each part as a single task. Obviously, we can get K tasks. If A is blocked like that, the task that corresponds with the row i is T^I.

If we block matrix $A_{q\times q}$ into three equal parts, we obtain3 tasks, as the following Fig. 1.:

$$\begin{bmatrix}
* & & & & & & & & \\
1 & * & & & & & & & \\
1 & 1 & * & & & & & & \\
* & * & * & * & & & & & \\
1 & 2 & 3 & 4 & * & & & & \\
1 & 1 & 2 & 3 & 4 & * & & & \\
* & * & * & * & * & * & * & & \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & * & \\
1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & *
\end{bmatrix}$$

Fig. 1. The parallel steps of the QR decomposition of the matrix $A_{q\times q}$ according to the block decomposition.

According to the definition of the plane revolving changing, we can know that during the QR changing of the matrix A, the expunged elements of the same row should be expunged in the line sequence. At the same time, if the (p,q) element is to be expunged, the front $q-1$ elements should be 0 of the k , p rows that required participating in the change, but the q elements should not be 0. So among the child tasks that obtained during the task partition, there are many child tasks that simultaneously satisfy the above conditions in each parallel step, so they can work parallelly. This is the blocking parallel of the plane revolving changing in the cluster system that pointed in this paper. Find out the child tasks that satisfy the above conditions in each parallel step, and distribute them to each processor to expunge zero. The parallel steps are $n-\frac{n}{k}+1$, they are concretely in Fig. 1., the figures in it are parallel steps.

4. THE ANALYSIS OF THE ACCELERATING RATIO AND THE EFFICIENT

If the matrix A is $n\times n$, we divided it into P equal parts, then we can obtain P child tasks, assume we have P processors, and the time of once arithmetic computation.(adding, minus, multiplication) or extraction is speeding t_t, t_2 ,we use T and T_p separately as the time of the serial implement of the p processors.

There is the following analysis between the serial and parallel computation of the plane revolving changing QR decomposition：

（1）Serial algorithm

Each plane revolving changing can define c , s , though one extraction and five times of arithmetic computation, and the $n-q$ changes that act on the q line ask for $6n$ times of arithmetic computation, so the total time of serial computation is:

$$\left[\sum_{j=1}^{n-1}(n-j)(6n+5)\right]\cdot t_1+\left(\sum_{j=1}^{n-1}(n-j)\right)\cdot t_2=\frac{n(n+1)(6n+5)}{2}t_1+\frac{n(n+1)}{2}t_2$$

In each step of the changing, in order to get Q^T , we use the changing matrix $Q_{p,q}$ to multiply the former Q^T , $Q_{p,q}$ acts on two rows of Q^T ,so each computation of the Q^T , ,there are 6n times of arithmetic computation, so the total serial time of computing Q^T is : $\frac{6n^2(n+1)}{2}t_2$.

So the total time is:

$$T_1=\frac{n(n+1)(12n+5)}{2}\cdot t_1+\frac{n(n+1)}{2}\cdot t_2$$

（2）Parallel algorithm

For the blocking parallel algorithm that design, we can only estimate its acceleration ratio and efficiency roughly.

The element number is $n-\frac{n}{k}+1$, parallel steps of each child task T^i $(1\le i\le P)$ in the blocking parallel algorithm are as follows:

$\frac{n}{2p}\left(\frac{n}{p}-1\right)$ （in the first parallel step）

$\frac{n}{p}-1$ （in other parallel steps）

This is also the time of the plane revolving changing in each parallel step. For each changing, in order to simplify the analysis, assure each time we compute all the elements of the rows that participated in the changing, as the analysis method used in the (1), we get the time of parallel computing:

$$T_2=\left[\frac{n}{2P}\left(\frac{n}{P}-1\right)+\left(n-\frac{n}{P}\right)\left(\frac{n}{P}-1\right)\right]\cdot\left[(12n+5)t_1+t_2\right]$$

This formula contains the computation of Q^T. This is because the changing matrix acting on the same rows of the R and Q^T, each time passing data of R to child tasks, it can pass the corresponding row of Q^T to the child task, then compute matrix R and Q^T simultaneously.

Through T_1 and T_2, we get the acceleration ratio and the efficient of the parallel algorithm, that is $s = \frac{T_1}{T_2}, E = \frac{s}{p}$. We must pay attention that the acceleration is of level $o(p)$, When $n \to \infty$, P increases, S also increases. When n is fixed, P increases, S will also increase at the first, but when P increases to a certain level, the spending in communication is of dominant status, so S will decrease. This is because as the increase of the processors, it leads to the decreasing of the partitioning granularity, and then the spending in communication will increase(please confer literatures[6,7]).

5. SIMULATION OF THE ALGORITHM

Use MPI to simulate the above algorithm in the internet of 1000Mbps, choose the equity model configuration, and realize it using Fortran .in the 8-hodes cluster system, we separately use 2 nodes, 4 nocks and 8 nodes to simulate the parallel algorithm of QRdecomposition. Each node is p_4 2.6GHZ. Assure the child takes of the QR decomposition is separately 2, 4 and 8.

The outcome of experiment is as follows.

Table 1. The outcome of the QR decomposition

n	Serial time (s)	K=2 P=2		
		Parallel time (s)	Acceleration ratio	Efficiency
800	115.63	83.79	1.38	0.69
1200	203.45	145.32	1.40	0.70
1600	340.87	250.64	1.36	0.68
n	Serial time (s)	K=4 P=4		
		Parallel time (s)	Acceleration ratio	Efficiency
800	115.63	73.18	1.58	0.79
1200	203.45	127.96	1.59	0.80
1600	340.87	214.38	1.59	0.80
n	Serial time (s)	K=8 P=8		
		Parallel time (s)	Acceleration ratio	Efficiency
800	115.63	75.08	1.54	0.77
1200	203.45	130.42	1.56	0.78
1600	340.87	219.92	1.55	0.77

n expresses the rank of the matrix in it.

We can see from the table that under the loom cluster environment, when p=2, the parallel algorithm gets a certain acceleration will also increase, but the acceleration ratio when k=4 is less than the acceleration ratio when k=8. This is because when k increases, the date that transported also increase, so it causes the increasing of the communication. This is fit for the former analysis.

6. CONCLUSION

The parallel algorithm of QR decomposition put forward in this text has the traits of little communication and high-level parallel degree. From the theoretical analysis and the experiment, we can say that it is fit for computing under the cluster environment.

7. REFERENCES

[1] A.M.Yang, C.F.Liu, "The Parallel Algorithm of GMRES (m) and Its Realization", 32nd Journal in 2005 of Computer Science, 2005, pp.309-313.

[2] J.F. Zhang, H.D. Jiang, "Direct Blocking Parallel Algorithm of Large Scale BF Equation Sets" Application Mechanics Ttransaction 2003, 20(4), pp.129-133.

[3] W. Michacl. "The Research and Development of Parallel Computation", Parallel Theory and Practice, 2000.

[4] X.M. Li, R.Z. Jiang, "Parallel Algorithm" Science and Technology Press, Changsha, Hunan, 1992.

[5] Y. Zhang, "Distributed Parallel Algorithm Designing, Analysis and Realization" Doctor Thesis of Electronic Science and Technology University 2001.

[6] Y.M. Chen, A.M. Yang, "MPI Parallel Programming Environment and Programming Research", 3rd Journal in 2005 of Academy Newspaper of Heibei Polytechnic University, 2005, pp. 41-44.

[7] Z.H. Du, "High-performance MPI programming Technology", Tinghua University Press, August of 2001.

Parallel algorithm to Analyze Failure process Of rock embodied cavity

ZeZhong Jiang
College of Civil Engineering, Southwest Jiaotong University
Chengdu, 610031, China
Email: qmrjzz@yahoo.com.cn

And

Tao Xie
[1]College of Civil Engineering, Southwest Jiaotong University
Chengdu, 610031, China
[2]College of Civil and Architecture Engineering, Guizhou University
Guiyang, 550022, China
Email: xie_tao96@163.com

ABSTRACT

Using high performance computer and a parallel processing algorithm method, this paper introduces finite element method in simulating the failure process of rock embodied cavity under monaxial compression. Powerful and accurate computation is becoming one of the most important tools for rock engineering problems, which demands supercomputers or large-scale parallel computer and related software that can be run on them. A parallel model of rock failure process analysis is proposed. The parallel algorithm has been completed using parallel domain decomposition method and discontinuous finite element method. The calculating program and its corresponding parallel algorithm are numerically stable, precise and quickly convergent. The numerical test shows that the algorithm can be used to simulate the failure process of rock and provide data and references to mechanical analyzing of rock.

Keywords: Parallel Computation, Rock Mechanics, Rock Failure Process, Finite Element Method, Domain Decomposition

1. INTRODUCTION

At the present time, parallel computation technique is been widely applied to every aspect of e — ngineering computation, such as ocean, aviation and heat energy. Computational rock mechanics is one of the furthest active fields of applying parallel computation technique. Rock mechanics problems are mostly the problems of rock failure. The rock failure is the process from local damage to macroscopical breakage [1, 2]. The study of rock failure process is a key research aspect of rock mechanics.

Applying numerical test method, P. Kryst[3] studied the three-dimensional crack propagation behavior of uniformity material. G. Lilliu[4] constituted numerical model of concrete failure under three-dimensions and heterogeneous condition. Chen yongqiang[5] studied the failure process of three-dimensional and heterogeneous brittle materials through numerical simulating. Whereas, because of applying the computation method of serial instruction flow of single computer, the scale and speed of computation is limited by the operation speed and EMS memory of computer. Powerful and accurate computation is becoming one of the most important tools for rock engineering problems. It demands supercomputers or large-scale parallel computer and related software that can be run on them.

A parallel model is proposed to analyze the rock failure process. The numerical test shows that the algorithm can be used to simulate the failure process of rock. The calculating program and its corresponding parallel algorithm are numerically stable, precise and quickly convergent.

2. DESCRIPTION OF FRACTURE MODELING

Geomaterials are usually treated as ideal CHILE (continuous, homogeneous, isotropic and linearly elastic) materials for the purpose of modeling. This treatment has been shown adequately in predicting the essential characteristics of geomaterials in a variety of simple applications. However, natural geomaterials, especially those rocks that incorporate fracturing at a variety of scales and hierarchies, are usually strongly discontinuous, inhomogeneous, anisotropic and nonelastic[6] and are poorly represented by the CHILE approximation.

In the fracture model, the material is analyzed at mesoscopic level. At the beginning, the element is considered having elastic properties that can be defined by Young's modulus and Poisson's ratio. The stress-strain curve of element is considered linear elastic until the given damage threshold is attained, and then is followed by softening. We choose the maximum tensile strain criterion and Mohr-Coulomb criterion respectively as the damage threshold. At any event, the tensile strain criterion is preferential.

In elastic damage mechanics, the elastic modulus of element may degrade gradually as damage progresses, the elastic modulus of damaged material is defined as follows.

(1)

Where ω represents the damage variable, E_0 and E are elastic modulus of the undamaged and the damaged material, respectively. Here the element as well as its damage is assumed isotropic and elastic, so the E, E_0 and ω are all scalar.

When element is under multi-axial stress state and its strength satisfies the Mohr-Coulomb criterion, the damage occurs, and we must consider the effect of other principal stress in this model during damage evolution process. When the Mohr-Coulomb criterion is met, we can calculate the maximum principal strain at the peak value of maximum principal stress (maximum compressive principal

stress).

(2)

In this respect, we assume that the shear damage evolution is only related to the maximum compressive principal strain ( ). So, the similar expression for damage variable, ω, can be extended to triaxial stress states for shear damage:

(3)

Where is also residual strength coefficient.

In the model, the specified displacement (or load) is applied on the specimen step by step. If some elements damage at this step, from the above derivation of damage variable, ω, as well as the Equation (1), the damaged elastic modulus of elements at each stress or strain level can be calculated when their stresses or strains meet one of the above two damage thresholds. Then calculation must be restarted under current boundary and loading conditions to achieve the stress redistribution in the specimen until no new damage occurs.

3. REALIZATION OF PARALLEL ALGORITHM

3.1. Circumstance of Parallel Computation

The circumstance of parallel computation is mainly made up of shareware EMS memory and distributed parallel system. At present, most large-scale parallel computer systems are the combine of them. In this study, the cluster of workstation (COW) is adopted as the form of parallel computer systems. The COW has many merits such as abundant network connections mode, convenient programmer, and favorable expansion [7].

The operating system of parallel computation is the fashionable Linux nowadays. Linux is famous as it's agile and high-efficiency. Because Linux impropriates few resource of computer system, the computing efficiency of Linux is far higher than Windows. The speed and stable of network communication of Linux is also better than Windows, which is very important for parallel computation.

The communications of node computer adopted the international standard which was the message passing interface (MPI). It is apt to use and has transplantable. The function of asynchronous of MPI is self-contained. It has formal and accurate definitions. Workload is widely cut down in parallel programming because that the MPI made use of the two basic function operations of BLAS and LAPACK.

3.2. Strategy of Parallel Computation

In the parallel computation, domain decomposition method is adopted which is one of furthest key problems. Domain decomposition disassembled complex system into simple sub-matter, so it is in favor of parallel computation. The base method of domain decomposition is that it disassembles complex system into simple system according to definite principle, and then distributes simple system to node computer for computing. The data commutation of node computer is by MPI.

If the amount of mutual grid of border area is decreased in domain decomposition, the data commutation quantum of parallel computation can been reduced and the efficiency of parallel computation can been heightened. Generally speaking, domain decomposition should satisfy these requests as follow: adaptive to anomalous computation area, same computation amount of each area and the least quantum of data commutation.

The illustration of parallel domain decomposition is showed in fig.1 as follows.

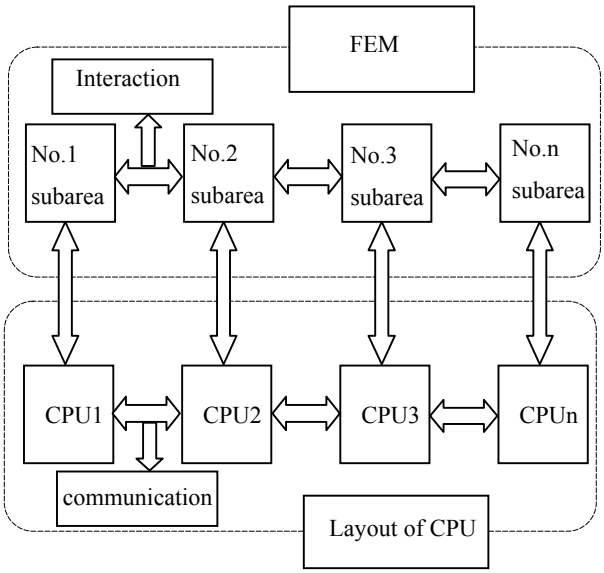


Fig. 1. Illustration of parallel domain decomposition

3.3. Parallel Algorithm of Linear Equation

In the solution of linear equation of parallel computation, the pretreatment conjugate grads method is adopted. The solution of linear equation is expresses as follows:

1). evaluating the initial value of , then calculating the value of and according to , and ;

2). for , calculating iterative computation.

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

3).if , the iterative computation is stopped.

4. FAILURE PROCESS ANALYSIS OF ROCK EMBODIED CAVITY

4.1. Parameter of Computation Model

There are three groups of specimens which embodied circinal cavity in the centre of specimens. The dimensions of specimens are 24mm×10mm×10mm. the value of radius of circinal cavity is respectively 0.75mm, 0.10mm and 1.25mm.

In the model, the specified displacement is applied on the specimen step by step along long axis. The mechanics parameter of the specimens is shown in the table 1.

Table 1. The mechanics parameter of the specimens

parameter	value	parameter	value
σ_c	180	E	90GPa
	15.5		1.43 MPam$^{1/2}$
	38°		2500kg/m^3

4.2. Speedups of Parallel Computation

The speedups are one of common estimate index in parallel computing. The result of speedups is shown in the table 2.

Table 2. Speedups of parallel computation

number of node	number of CPU	time of computing	speedups
1	2	140.50	1.00
2	4	70.19	2.00
3	6	53.90	2.61
4	8	41.03	3.42
5	10	33.85	4.15

In the table 2, the value of speedups is relative to the computation time of 2 CPU.

4.3. Result of Parallel Computation

The result is shown in the fig.2 and the table 3 as follows.

Table 3. The Peak Strength of specimens

	0.75mm	$r=1.0$mm	$r=1.25$mm
Experimental results	145MPa	132MPa	78MPa
Parallel computing results	128MPa	107MPa	91MPa
Analytical results	139MPa	69MPa	35MPa

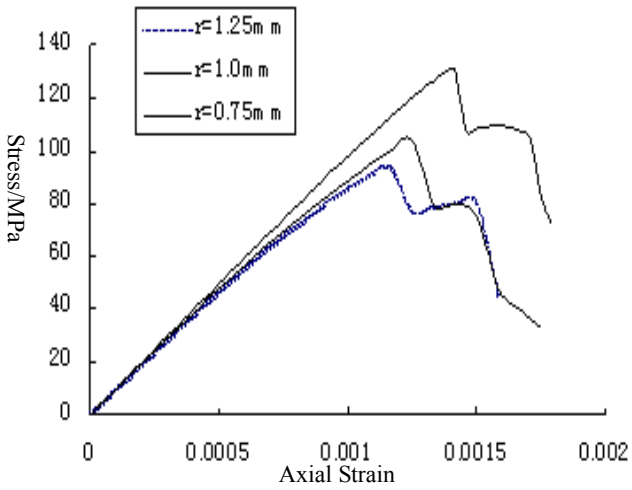


Fig.2 The curve of stress-strain

The failure process of rock embodied circinal cavity is shown in the fig.3 as follows.

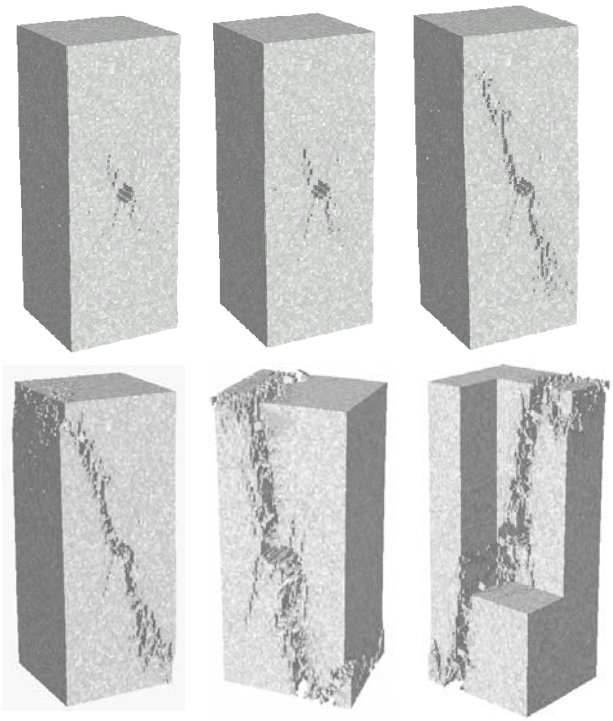


Fig. 3. The failure process of specimens

4.4. Computation Result Analysis

From the results of parallel computation of rock embodied circinal cavity, we can find some rule as follow. Under the condition that the value of is small, the value of stress of initial crack propagation is big and the speed of crack propagation is rapid when the crack propagated initially.

The crack burgeoned along the cavity, but the direction of crack propagation departure from the direction of principal stress and along the direction of . Furthermore, the main force of crack propagation is changed from tense to cut, and the specimens have the character of

cut breakage.

Under the condition that the value of is big, the state of crack propagation is stable. The crack initiated along the direction of principal stress and the value of stress of initial crack propagation is small. The peak load decreased as the value of , and the main force of crack propagation is tensile stress. The crack propagation path follows the direction of principal tensile stress. The specimens have the character of tense breakage.

5. CONCLUSIONS

This paper presents the results of a study aimed at the parallel computation of the failure process of rock embodied cavity under monaxial compression. It shows that the peak load decreases as the value of increases, the crack propagation path follows some regularity in general and the main force of crack propagation is still tensile stress.

The parallel algorithm has been completed using parallel domain decomposition method and discontinuous finite element method. The calculating program and its corresponding parallel algorithm are numerically stable, precise and quickly convergent. The numerical test shows that the algorithm can be used to simulate the failure process of rock and provide data and references to mechanical analyzing of rock.

6. REFERENCES

[1] C.A. Tang, Numerical modelling of the influence of heterogeneity on 3D polygonal fracture in layered materials. Failure Mechanics Letters (www.rfpa.cn/fml), Vol.1 (2005):1-2

[2] C.A. Tang, Numerical investigation of particle breakage as applied to mechanical crushing-part 1: Single-particle Breakage. International Journal of Rock Mechanics & Mining Sciences, 2001, 38:1147-1162.

[3] P. Kryst, T. Belytschko, The element free Galerkin method for dynamic propagation of arbitrary 3D cracks [J]. International Journal for Numerical Methods in Engineering, 1999, 44(6):767–800.

[4] G, Lilliu, 3D lattice type fracture model for concrete [J]. Engineering Fracture Mechanics, 2003, 70(7/8):927–941.

[5] Y.Q. Chen, X.P. Zheng, et al. Numerical simulation of failure processes in 3D heterogeneous brittle material [J]. Acta Mechanica Sinica，2002,34(3): 351–361. (in Chinese))

[6] J. A. Hudson, J. P. Harrison, An Introduction to the Principles. Engineering Rock Mechanics, 1997

[7] G.L. Chen, Parallel Computation [M]. Beijing:Higher Education Publishing House, 1999.(in Chinese)

A Parallel Computing Procedure for Lower and Upper Bounds on Linear Functionals of Solutions to PDE: Appliction to Stress Intensity Factors in Bimaterials

Zhaocheng Xuan
Department of Computer Science, Tianjin University of Technology and Education,
Tianjin 300222, China
Email: xuanzc@tute.edu.cn

ABSTRACT

A computing procedure for the bounds on stress intensity factors in bimaterial is presented based on the a-posteriori error estimate in output of finite element solutions. The stress intensity factors on coarse finite element mesh are first obtained by two points extrapolation method with the displacements of two nodes (or node pairs) on the crack edges computed by finite element method. Then a-posteriori bounding procedure based on the finite element error estimate is utilized to compute the bounds to stress intensity factors. The computation of the error estimate is performed by solving independent elemental Neumann subproblems decomposed from the finite element model, thus the computing procedure is parallel and potential to solve large scale structural problems. The lower and upper bounds on stress intensity factors of a bimaterial interface crack problem are computed with the present method to verify the procedure.

Keywords: Bounds, Finite elements, SIF.

1. INTRODUCTION

The quantities of interest in the civil and mechanical engineering design are usually the linear or nonlinear functionals of displacements of structure, e.g. stresses, stress intensity factors (SIFs), which are usually referred to as the outputs of the displacements. The estimate on the output is often more important than only on the finite element solution in engineering viewpoint. To measure the quality of numerical results for these locally defined outputs some approaches for local error estimators and related strategies for finite element mesh refinement have been introduced in recent years [1,6].

The SIF is used in the study of brittle fracture, fatigue, stress corrosion cracking, and to some extend for creep crack growth. Since the analytical methods, such as integral methods, for calculating the SIFs are only limited tosome simple cases of fracture mechanics problems, the finite element method are widely used as the alternatives to treat the complicated cases. Finite element analysis of linear elastic fracture mechanics problems is complicated by the presence of the singular and finite non-singular stress distributions in the crack tip region. One big problem with finite element analysis is verification of the finite element solution. This motivated the establishment of the bound, or estimate, or measure, on the error between the numerical results and the exact solutions of the mathematical model [4].

This paper focuses on computing the bounds on SIF. The aim is not to compute the SIFs, but to compute the bounds to the SIFs on the fine mesh with the results obtained from the coarse mesh. In finite element analysis, it is well known that the results obtained from the coarse mesh are less accurate, but the cost is less expensive, while the results from the fine mesh are more accurate but the cost is more expensive. Thus it is useful in practice if we could give the bounds to the SIFs on the fine mesh only with the cost of calculating local problems relating to the coarse mesh.

2. A PARALLEL OUTPUT BOUND PROCEDURE FOR THE LINEAR FUNCTIONAL OF FINITE ELEMENT SOLUTIONS

An elastic body with a bounded region $\Omega \subset \Re^2$ is considered in this paper for developing the parallel computing procedure for output bounds. The boundary of Ω is assumed piecewise smooth, and composed of Dirichlet portion Γ_D and Neumann portion Γ_N, i.e. $\partial\Omega = \Gamma_D \cup \Gamma_N$. As usual, $u : \Omega \to \Re^2$ denotes the sought displacement, f is the load, g is the prescribed boundary traction, and n denotes the unit outward normal. The stress tensor $\sigma(u)$ is related to the strain tensor, $\varepsilon(u) = \frac{1}{2}(\nabla + \nabla^T)\cdot u$, by the material law, $\sigma(u) = E \cdot \varepsilon(u)$, in which E is the elastic moduli. The weak form of the elasticity equation is to find u in X such that

$$a(u,v) = (f,v) + \langle g, v\rangle, \quad \forall v \in H^1(\Omega), \tag{1}$$

in which $a(u,v) = \int_\Omega \sigma(u) : \varepsilon(v)\mathrm{d}\Omega$, $(f,v) = \int_\Omega fv\mathrm{d}\Omega$, $\langle g, v\rangle = \int_{\Gamma_N} gv\mathrm{d}s$, and $X = \{v \in H^1(\Omega) \mid v = 0 \ on \ \Gamma_D\}$.

$H^k(\Omega)$ denotes the usual Sobolev space. In this paper, the energy norm associated with the bilinear form $a(\cdot,\cdot)$ is uesed in this paper, which is defined as $|||u|||^2 = a(u,u)$.

In order to obtain the approximate solutions of the weak problem, a finite dimensional counterpart of all these variational forms above can be built using the Galerkin finite element method. Now we define X_H to be the piecewise linear continuous finiteelement space associated with the triangulation T_H of the working or coarse H-mesh, T_H, consisting of N_H elements T_H, $X_H = \{v \in X \mid v|_{T_H} \in P_1(T_H), \forall T_H \in T_H\}$, in which $P_1(T_H)$ denotes the space of linear polynomials over T_H. The corresponding finite element solution to X_H is found as u_H, then the discretization error will be $e^{pr} = u - u_H$.

Generally we can not find the exact solution u, but when the finite element mesh is generated so fine that the finite element solution u_h is almost the exact solution.

Let X_h be the function space with such kind of fine or *truth* finite element mesh T_h consisting of N_h elements T_h, $X_h = \{v \in X \mid v|_{T_h} \in P_1(T_h), \forall T_h \in T_h\}$, in which $P_1(T_h)$ is the space of linear polynomials over T_h. It is obvious that $X_H \subset X_h \subset X$. We also require that a fine mesh T_h be only within one coarse element T_H. The error e^{pr} can be written as $e^{pr} = u - u_H = u - u_h + u_h - u_H \approx u_h - u_H$, because the error between u and u_h can be neglected due to the very fine mesh T_h, here u_h is the finite element solution to X_h.

Now let us consider the output which is the linear functional of the solution u defined to be $s = l^O(u)$, i.e. $l^O : H^1(\Omega) \mapsto \Re$, then on the working mesh the output, s_H, is $s_H = l^O(u_H)$, while the *exact* output is $s_h = l^O(u_h)$, then the error of output is $s_h - s_H = l^O(u_h) - l^O(u_H) = l^O(e^{pr})$, and therefore $s_h = s_H + l^O(e^{pr})$.

We can see the output corresponding to fine mesh T_h is related to the output obtained from coarse mesh and the output of error of finite element solution. Since our end is to calculate the finite element error bound to s_h using the information obtained from u_H, s_H, we will not really calculate the output s_h on the fine mesh T_h, but we will use the domain decomposition technique to take each working mesh T_H as a subdomain to calculate the output bound to s_h on T_H independently. Now we introduce the *broken* spaces

$$\hat{X}_H = \{v \in L_2(\Omega) \mid v|_{T_H} \in P_1(T_H), \quad \forall T_H \in T_H\},$$
$$\hat{X}_h = \{v \in \hat{X}_H \mid v|_{T_h} \in P_1(T_h), \quad \forall T_h \in T_h\},$$

where $L_2(\Omega)$ is the square integrable functional space. The defined finite element mesh and space are illustrated in Fig. 1.

Let E_I represent the coarse element edges in the interior of Ω, here we assume that boundary $\partial\Omega$ consist of the edges of $T_H \in T_H$. Let N_{T_H} denote all neighbor elements sharing common edges ∂T_H with the coarse element T_H, i.e. $N_{T_H} = \left\{T_H' \in T_H \mid T_H' \cap T_H = E_I\right\}$. Let $E(T_H)$ and $E(T_h)$ denote the set of open edges in the triangulation T_H and T_h, respectively. Then according to the above assumption, for coarse mesh we have $E(T_H) = E_I \cup \Gamma_D \cup \Gamma_N$.

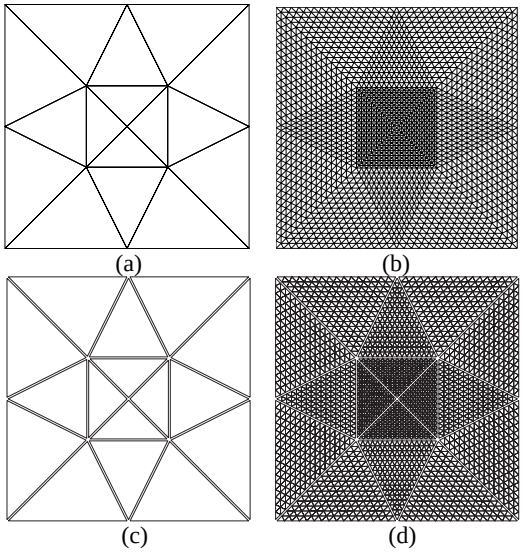


Fig. 1. (a) the coarse mesh T_H and associated finite element space X_H as defined; (b) the fine mesh T_h and associated finite element space X_h; (c) the broken coarse mesh and finite element space $\hat{X}_H$; (d) the broken fine mesh and the finite element space $\hat{X}_h$.

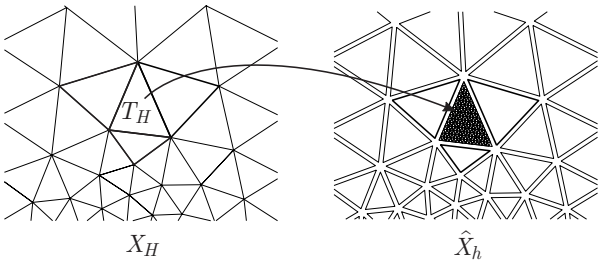


Fig. 2. In the procedure for computing the output bounds, an individual computing for the bounds to the energy norm of the error between the finite element solutions obtained on X_H and $\hat{X}_h$ will be performed on each coarse element T_H in working mesh T_H, the data of the coarse element and its neighbor elements obtained in both primal and dual problems will be used to calculate the error bounds to the finite element solutions.

The summary of the output bounds procedure is reorganized as the following:

STEP 1, solve the primal problem

$$a(u_H, w) = (f, w) + \langle g, w \rangle, \quad \forall w \in X_H$$

on coarse mesh to find $u_H \in X_H$, then we can obtain the working output, s_H, as $s_H = l^O(u_H)$. The residual is defined as $R^{pr}(w) = (f, w) + \langle g, w \rangle - a(u_H, w)$.

STEP 2, solve the dual problem

$$a(w, \psi_H) = -l^O(w), \quad \forall w \in X_H$$

on coarse mesh to find $\psi_H \in X_H$, and the residual is defined as $R^{du}(w) = -l^O(w) - a(w, \psi_H)$.

STEP 3, solve $2 \times N_H$ subproblems (2) and (3) on each coarse element to obtain $\hat{e}^{pr}$ and $\hat{e}^{du}$.

$$a_{T_H}(\hat{e}^{pr}, w) = R^{pr}_{T_H}(w - I_H w) +$$

$$\left\langle n_{T_H}\cdot\overline{\sigma}(u_H), w - I_H w\right\rangle_{\partial T_H\cap E_I},\quad \forall w\in\hat{X}_h, \tag{2}$$

$$a_{T_H}(\hat{e}^{du}, w) = R^{du}_{T_H}(w - I_H w) + \left\langle n_{T_H}\cdot\overline{\sigma}(\psi_H), w - I_H w\right\rangle_{\partial T_H\cap E_I},\quad \forall w\in\hat{X}_h, \tag{3}$$

where $I_H:\hat{X}_h\mapsto\hat{X}_H$ denotes a local polynomial interpolation operator such that $I_H v\in\hat{X}_H$ for both $v\in\hat{X}_h$ and $v\in\hat{X}_H$. n_{T_H} is the unit outward normal of element T_H, $\overline{\sigma}(u_H)$ represents the average stress on the common edges of working element T_H and its neighbor elements in primal problem, refer to Fig. 1.,

$$\overline{\sigma}(u_H) = \frac{1}{2}\left(\sigma(u_H)\big|^{+}_{\partial T_H} + \sigma(u_H)\big|^{-}_{\partial T_H}\right). \tag{4}$$

STEP 4, compute the lower and upper output bounds, $s_{LO}(T_H)$ and $s_{UP}(T_H)$, as

$$s_{LO}(T_H) = s_H - \frac{1}{2}a(\hat{e}^{pr},\hat{e}^{du}) - \frac{1}{2}|||\,\hat{e}^{pr}\,|||\,|||\,\hat{e}^{du}\,|||, \tag{5}$$

$$s_{UP}(T_H) = s_H - \frac{1}{2}a(\hat{e}^{pr},\hat{e}^{du}) + \frac{1}{2}|||\,\hat{e}^{pr}\,|||\,|||\,\hat{e}^{du}\,|||, \tag{6}$$

where

$$a(\hat{e}^{pr},\hat{e}^{du}) = \sum_{T_H\in T_H} a_{T_H}(\hat{e}^{pr},\hat{e}^{du}),$$

$$|||\,\hat{e}^{pr}\,||| = \left(\sum_{T_H\in T_H} a_{T_H}(\hat{e}^{pr},\hat{e}^{pr})\right)^{1/2},$$

$$|||\,\hat{e}^{du}\,||| = \left(\sum_{T_H\in T_H} a_{T_H}(\hat{e}^{du},\hat{e}^{du})\right)^{1/2}.$$

In this paper, we will take the SIFs in fracture mechanics as the output and calculate the bounds to the SIFs by finite element analysis. In the above procedure, the output is the linear functional of solutions of partial differential equation. Fortunately, the SIF is the linear functional of the solutions of elasticity equation, but in the interface crack of bimaterials, the linear formulations of each separated SIF can not be easily expressed as in the crack problem of homogeneous material. In the next section, we will derive the linear formulations, the two points displacements extrapolation formulations, for the SIFs in both homogeneous and bimaterial problems.

3. COMPUTING SIFS OF INTERFACE CRACK IN BIMATERIALS

Let us consider an interface crack lying in the bond line between two homogeneous isotropic elastic materials as shown in Fig. 2. The linear elastic solution in the crack tip region is developed using the bimaterial constant ε given by

$$\varepsilon = \frac{1}{2\pi}\ln\left(\frac{1-\beta}{1+\beta}\right),$$

with β being one of the Dundurs' parameters defined as

$$\beta = \frac{(\kappa_2-1)\frac{\mu_1}{\mu_2} - (\kappa_1-1)}{(\kappa_2+1)\frac{\mu_1}{\mu_2} + (\kappa_1+1)},$$

where the constant κ_j is introduced, such that $\kappa_j = 3-4\nu_j$ for plane strain and $\kappa_j = (3-\nu_j)/(1+\nu_j)$ for plane stress, for $j = 1,2$.

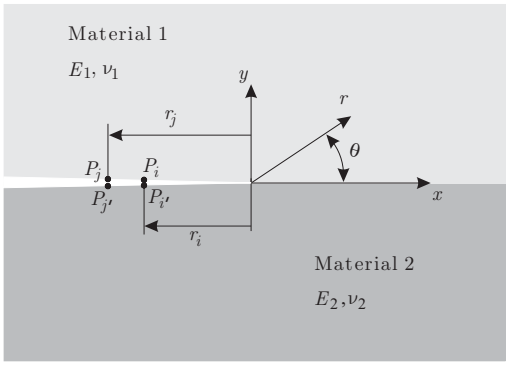


Fig. 3. Geometry and coordinate definitions for an interface crack.

The bimaterial cracks is always intrinsically mixed mode regardless of the nature of remote loading conditions. A complex SIF is defined as $K \equiv K_1 + iK_2 = |K|e^{i\vartheta}$, here ϑ is the phase angle, it reduces to $K = K_I + iK_{II}$ if the material mismatch is zero, where K_I and K_{II} are SIFs associated with pure mode-I and mode-II loading. The stresses, σ_{yy} and σ_{xy}, at the interface directly ahead of the tip, at $\theta = 0$, aregiven by

$$\sigma \equiv \sigma_{yy} + i\sigma_{xy} = \frac{K_1 + iK_2}{\sqrt{2\pi r}}\left(\frac{r}{2a}\right)^{i\varepsilon}.$$

Then the stress can be expressed, from the above equation, as

$$\sigma_{yy} = \frac{1}{\sqrt{2\pi r}}\left(K_1\cos\varphi - K_2\sin\varphi\right), \tag{7}$$

$$\sigma_{xy} = \frac{1}{\sqrt{2\pi r}}\left(K_1\sin\varphi + K_2\cos\varphi\right), \tag{8}$$

in which $\varphi = \varepsilon\ln\left(\frac{r}{2a}\right)$. From the stress equations, Malyshev and Salganik[7] showed that the energy release rate, G, in terms of the complex intensity factor is

$$G = \frac{1}{16\cosh^2(\varepsilon\pi)}\left(\frac{\kappa_1+1}{\mu_1} + \frac{\kappa_2+1}{\mu_2}\right)K_0^2, \tag{9}$$

in which $K_0 = \sqrt{K_1^2 + K_2^2}$. We can see the energy release rate is proportional to K_0, thus it is important to calculate the error bound to K_0 in finite element analysis of crack between dissimilar materials. In the displacement formulations, the associated crack surface displacements, δ_y and δ_x, at the place with a distance r behind the tip are given as

$$\begin{aligned}\delta &\equiv \delta_y + i\delta_x \\ &= (u_y + iu_x)_{\theta=\pi} - (u_y + iu_x)_{\theta=-\pi} \\ &= \frac{K_1 + iK_2}{2(1+2i\varepsilon)\cosh(\varepsilon\pi)}\left(\frac{\kappa_1+1}{\mu_1} + \frac{\kappa_2+1}{\mu_2}\right)\sqrt{\frac{r}{2\pi}}\left(\frac{r}{2a}\right)^{i\varepsilon}.\end{aligned} \tag{10}$$

The displacements formulations were originally proposed by Hutchinson et al [8]. δ_y and δ_x are the opening and sliding displacement jumps across the crack faces. A displacement extrapolation technique can be developed using the complex crack opening displacement to compute the complex SIF in bimaterial specimens. Considering only the K-domain terms in the series solution for the bimaterial crack, from (10) the relative displacements of the

crack faces can be given as

$$\delta_x(r) = \eta\sqrt{\frac{r}{2\pi}}\left(\Psi_1(\varphi)K_1 + \Psi_2(\varphi)K_2\right), \tag{11}$$

$$\delta_y(r) = \eta\sqrt{\frac{r}{2\pi}}\left(\Psi_2(\varphi)K_1 - \Psi_1(\varphi)K_2\right), \tag{12}$$

where

$$\Psi_1(\varphi) = \sin(\varphi) - 2\varepsilon\cos(\varphi),$$
$$\Psi_2(\varphi) = \cos(\varphi) + 2\varepsilon\sin(\varphi),$$
$$\eta = \frac{1}{2(1+4\varepsilon^2)\cosh(\varepsilon\pi)}\left(\frac{\kappa_1+1}{\mu_1} + \frac{\kappa_2+1}{\mu_2}\right).$$

Due to the importance of the energy release rate, G, in the interfacial cracks, and according to the relationship between the energy release rate and the SIF K_0 in equation (9), we now focus on deriving the displacement formulations related to K_0,

$$K_0^* = \lim_{r\to 0}\left(K_0(r) = \frac{1}{\eta\sqrt{(1+4\varepsilon^2)r}}\sqrt{\delta_x^2(r) + \delta_y^2(r)}\right). \tag{13}$$

With two points extrapolation method the SIFs at the crack tip can be obtained from the results calculated from the displacements at the point pairs $i - i'$ and $j - j'$. The SIF K_0 at the crack tip is expressed as

$$K_0^* = \frac{r_j}{r_j - r_i}K_0(r_i) - \frac{r_i}{r_j - r_i}K_0(r_j). \tag{14}$$

From (13) we can see $K_0(r_i)$ and $K_0(r_j)$ are related to the nonlinear formulation of the relative displacements, $\sqrt{\delta_x^2(r_i) + \delta_y^2(r_i)}$ and $\sqrt{\delta_x^2(r_j) + \delta_y^2(r_j)}$. can be linearized at point r_i as The linearization of the nonlinear term is equivalent to changing the cone into a pyramid, see the cone, $\sqrt{\delta_x^2(r) + \delta_y^2(r)}$, and the pyramid. According to (13), rearranging (14) leads to the SIF K_0 at the crack tip in the formulation of displacements at two point pairs,

$$K_0^* = \frac{1}{\eta\sqrt{(1+4\varepsilon^2)}}\left(\frac{r_j}{(r_j - r_i)\sqrt{\left(\delta_x^2(r_i) + \delta_y^2(r_i)\right)r_i}}\Theta_i u_i - \frac{r_i}{(r_j - r_i)\sqrt{\left(\delta_x^2(r_j) + \delta_y^2(r_j)\right)r_j}}\Theta_j u_j\right), \tag{15}$$

in which

$$\Theta_i = (\delta_x(r_i), \delta_y(r_i), -\delta_x(r_i), -\delta_y(r_i));$$
$$\Theta_j = (\delta_x(r_j), \delta_y(r_j), -\delta_x(r_j), -\delta_y(r_j)).$$

For characterizing the behavior of the interface crack it is necessary to evaluate the values K_1 and K_2 separately. The ratio $K_r = K_2/K_1$ at the crack tip can be obtained by the displacement extrapolation method,

$$K_r^* = \lim_{r\to 0}\left\{K_r(r) = \frac{H_1(r)\delta_x(r) - H_2(r)\delta_y(r)}{H_2(r)\delta_x(r) + H_1(r)\delta_y(r)}\right\}, \tag{16}$$

where $H_1(r) = 1 + 2\varepsilon\,\mathrm{tg}\varphi$ and $H_2(r) = \mathrm{tg}\varphi - 2\varepsilon$.

Analogue to K_0 we can obtain the ratio expressed with δ_x and δ_y at the crack tip with two point extrapolation method,

$$K_r^* = \frac{r_j}{r_j - r_i}K_r(r_i) - \frac{r_i}{r_j - r_i}K_r(r_j). \tag{17}$$

With the same method used for K_0, the ratio can be linearized at $r = r_i$ as

$$K_r \approx K_r(r_i) + (1+4\varepsilon^2)\omega_i\left(\delta_y(r_i)\delta_x(r) - \delta_x(r_i)\delta_y(r)\right), \tag{18}$$

with

$$\omega_i = (1 + \mathrm{tg}^2\varphi(r_i))/\left(H_2(r_i)\delta_x(r_i) + H_1(r_i)\delta_y(r_i)\right)^2,$$

and at $r = r_j$ as

$$K_r \approx K_r(r_j) + (1+4\varepsilon^2)\omega_j\left(\delta_y(r_j)\delta_x(r) - \delta_x(r_j)\delta_y(r)\right), \tag{19}$$

with

$$\omega_j = (1 + \mathrm{tg}^2\varphi(r_j))/\left(H_2(r_j)\delta_x(r_j) + H_1(r_j)\delta_y(r_j)\right)^2.$$

It can be seen that for $r = r_i$ and $r = r_j$ the above two equations are respectively $K_r(r_i)$ and $K_r(r_j)$, therefore using the two equations, (18), (19), in places of $K_r(r_i)$ and $K_r(r_j)$ in (17) leads to

$$K_r^* \approx \frac{r_j}{r_j - r_i}K_r(r_i) - \frac{r_i}{r_j - r_i}K_r(r_j) + (1+4\varepsilon^2)\left(\frac{r_j\omega_i}{r_j - r_i}\left(\delta_y(r_i)\delta_x(r) - \delta_x(r_i)\delta_y(r)\right) - \frac{r_i\omega_j}{r_j - r_i}\left(\delta_y(r_j)\delta_x(r) - \delta_x(r_j)\delta_y(r)\right)\right). \tag{20}$$

Rearranging the above equation (20) with the same method as for K_0 results in

$$K_r^* \approx \frac{r_j}{r_j - r_i}K_r(r_i) - \frac{r_i}{r_j - r_i}K_r(r_j) + K_r^l, \tag{21}$$

in which

$$K_r^l = (1+4\varepsilon^2)\left(\frac{r_j\omega_i}{r_j - r_i}\Lambda_i u_i - \frac{r_i\omega_j}{r_j - r_i}\Lambda_j u_j\right), \tag{22}$$

with

$$\Lambda_i = (\delta_y(r_i), -\delta_x(r_i), -\delta_y(r_i), \delta_x(r_i));$$
$$\Lambda_j = (\delta_y(r_j), -\delta_x(r_j), -\delta_y(r_j), \delta_x(r_j)).$$

Comparing equations (17) and (21), we find K_r^l is approximately zero. Then we will first use (22) to compute the bounds to K_r^l, and then simply add (17) to obtain the bounds to K_r^*.

The SIFs, K_0^* in (15) and K_r^l in (22), have been formulated as the linear functions of the nodes displacements on the crack edges, they can be expressed as the general form

$$s = \mathbf{C}_i u_i - \mathbf{C}_j u_j, \tag{23}$$

where $\mathbf{C}_i$ and $\mathbf{C}_j$ are the terms corresponding to that in the equations (15) and (22). We relate the SIFs with the displacements at two nodes or node pairs on the crack edges. Since $u \in H^1(\Omega)$, the quantity related with the pointwise displacement is not bounded in two or three dimension

space. We use the local averages with mollifying kernels to circumvent this issue [3].

4. A BIMATERIAL EXAMPLE

A finite dissimilar plate with edge crack subjected to a uniform tensile stress as shown in Fig. 3. will be used as the example to verify the bounding procedure. The plate is assumed in the state of plane stress, and geometrical and physical properties are listed in the figure.

Since this example is an interfacial fracture problem, there are both open mode and shear mode SIFs, K_1 and K_2, which are much similar to the mixed mode homogeneous fracture problem. The SIFs, $K_0 = \sqrt{K_1^2 + K_2^2}$ and $K_r = K_2/K_1$, are plotted in Fig. 5. At the crack tip, $K_0^* = 4.1125$ and $K_r^* = -0.2481$. Compared with the results given by [9], $K_0 = 4.0850$, $K_r = -0.2470$, the relative differences are 0.67% and 0.45%, respectively.

The bounds results for the SIFs are listed in Table 1. for K_0 and Table 2. for K_r. We can see that they do give the lower and upper bounds to the SIFs. From the results of the ratios listed in the bottom of each table, the effectivity of the bounds are shown clearly. In Table 2., since the bounds are negative, the absolute value of K_r^* is used in the ratios for expressing the bounding property clearly. The convergence rate, which is about 1.2, of the bound gaps $s_{UP} - s_{LO}$ for the SIF K_0. The bounds results of the stress intensity factors on the fine mesh with mesh size, $H/16$, computed with different meshes with mesh sizes, H, $H/2$, $H/4$, $H/8$.

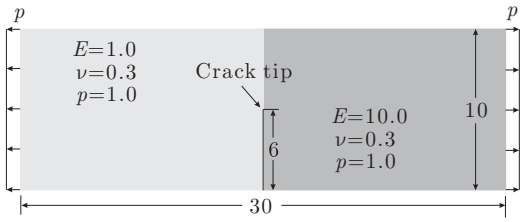


Fig. 4. An interface crack in bimaterial.

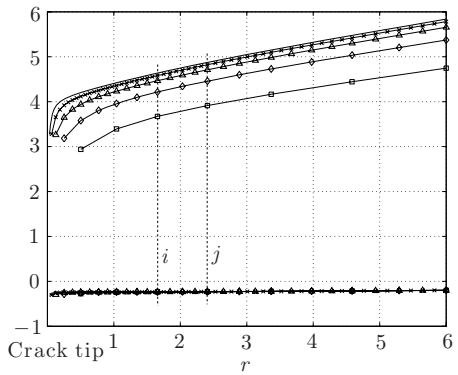


Fig. 5. The square root $K_0 = \sqrt{K_1^2 + K_2^2}$ (top) and ratio $K_r = K_2/K_1$ (down) vs r.

Table 1. Bounds results for K_0.

Mesh size	H	$\frac{1}{2}H$	$\frac{1}{4}H$	$\frac{1}{8}H$	$\frac{1}{16}H$
$s_H = K_0$	3.146	3.676	3.934	4.055	4.113
s_{LO}	-2.289	0.621	2.478	3.474	4.113
s_{UP}	13.151	8.428	5.943	4.786	4.113
s_{pre}	5.431	4.525	4.211	4.130	4.113
s_H/K_0^*	0.765	0.894	0.957	0.986	1.000
s_{LO}/K_0^*	-0.557	0.151	0.603	0.845	1.000
s_{UP}/K_0^*	3.198	2.049	1.445	1.164	1.000
s_{pre}/K_0^*	1.321	1.100	1.024	1.004	1.000

Table 2. Bounds results for K_r.

Mesh size	H	$\frac{1}{2}H$	$\frac{1}{4}H$	$\frac{1}{8}H$	$\frac{1}{16}H$
$s_H = K_r$	-0.243	-0.245	-0.246	-0.248	-0.248
s_{LO}	-0.853	-0.491	-0.345	-0.282	-0.248
s_{UP}	0.377	0.003	-0.150	-0.214	-0.248
s_{pre}	-0.238	-0.244	-0.247	-0.248	-0.248
$s_H/\lvert K_r^* \rvert$	-0.977	-0.989	-0.998	-1.000	-1.000
$s_{LO}/\lvert K_r^* \rvert$	-3.439	-1.979	-1.389	-1.138	-1.000
$s_{UP}/\lvert K_r^* \rvert$	1.517	0.013	-0.605	-0.861	-1.000
$s_{pre}/\lvert K_r^* \rvert$	-0.961	-0.983	-0.997	-1.000	-1.000

5. REFERENCES

[1] M. Paraschivoiu, J. Peraire, et al, A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations, Comput. Meth. App. Mech. Eng. 150, 1997, pp.289-312.

[2] J. Peraire, A.T. Patera, Bounds for linear-functional outputs of coercive partial differential equations: Local indicators and adaptive refinement. Proceedings of the Workshop on New Advances in Adaptive Computational Methods in Mechanics, edited by P. Ladeveze and J.T. Oden, Elsevier, 1998

[3] J.T. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method. Comput. & Math. App. 41, 2001, 735-756.

[4] M. Ainsworth, J.T. Oden, A posteriori error estimation in finite element analysis, John Wiley & Sons, New York, 2000.

[5] Z.C. Xuan, N. Parés, et al, Computing upper and lower bounds for the J-integral in two-dimensional linear elasticity, Comput. Meth. App. Mech. Eng. 195, 2006, 430-443.

[6] Z.C. Xuan, B.C. Khoo, et al, Computing bounds to mixed mode stress intensity factors in elasticity, Arch. App. Mech. 75, 2006, 193-205.

[7] N.M. Malyshev, R.L. Salganik, The adhesive joints using the theory of crack. Int. J. Fract. l, 1965, pp. 114-128.

[8] J.W. Hutchinson, M. Mear, et al, Crack paralleling an interface between dissimilar materials, ASME J. App. Mech. 54, 1987, 828-832.

[9] R. Yuuki, S.B. Cho, Efficient boundary element analysis

of stress intensity factors for intensive cracks in dissimilar materials, Eng. Fract. Mech. 9, 1977, 931-938

PARALLELIZING AN IMMUNE-INSPIRED ALGORITHM FOR FUZZY DATA CLUSTERING

Li Liu
School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, 214122, China
Email: yxhll@pub.wx.jsinfo.net

ABSTRACT

In recent years, there has been a growth of use of artificial immune network for data analysis. The huge amount of data generated by data intensive industries has pushed data analysis toward parallel implementation. The work in this paper presents the first step at realizing a parallel artificial immune network. A parallel version of artificial immune network for fuzzy data clustering was proposed to explore inherent distribution and parallelism of artificial immune system and to achieve efficiency for data analysis. Experimental results have shown that the proposed approach is simple and effective in decreasing overall time.

Keywords: artificial immune network, fuzzy data clustering, parallelization.

1. INTRODUCTION

Cluster analysis divides data into groups (clusters) such that similar data objects belong to the same cluster and dissimilar data objects to different clusters. In fuzzy clustering, Membership degrees between zero and one are used instead of crisp assignments of the data to clusters, so fuzzy clustering is much more suitable for real-world applications [1,2]. The objective function based fuzzy clustering algorithms are often used in clustering tasks. By minimizing the objective function, which measures the overall dissimilarity within clusters, we can obtain the optimal partition. Fuzzy C-Means algorithm, the most popular objective function based fuzzy clustering method, however, is a local optimization algorithm, a random selection of initial value may lead to crisp premise fuzzy set.

In recent years, there has been a growth of investigation into the use of the mammalian immune system as a source of inspiration and metaphor for computational tasks [3,4,5]. One object of this investigation has been the exploration of the global search and learning capabilities demonstrated by these biological systems. Based on the clone and affinity mutation principals of biological immunity mechanism, containing memory cells, the immune algorithm is capability of maintaining local optima solutions and exploring the global optima. The immune network imitates the activities of the immune cells, the emergence of memory and the discrimination between our own cells (known as self) and external invaders (known as non-self).

Another appealing aspect of biological immune systems is their inherent distribution and parallelism. Within the AIS community, there has been some exploration of the distributed nature of the immune system as evidenced in algorithms for network intrusion detection as well as some ideas for distributed robot control. Few works have been undertaken, to date, in the parallelization of immune-inspired algorithms. [6] [7] demonstrated that parallel techniques can be effectively applies to AIS. However, paralleling an artificial network has not been known before.

The algorithm presented in this paper is based upon previous works. It is a parallelized adaptation of an immune inspired fuzzy clustering algorithm, named as AINFCM, to study how to improve efficiency and exert inherent distribution and parallelization. The outline of this paper was as followings: section 2 gives a brief overview of immune inspired fuzzy clustering algorithm AINFCM; section 3 discusses parallelization aspects of algorithm; section 4 presents experimental results; section 5 gives some conclusion and future work aspects.

2. OVERVIEW OF AINFCM

We proposed the Artificial Immune Network for Fuzzy Clustering Method (AINFCM) [8], which solves the traditional fuzzy clustering problems by searching for the optimal centers of clusters using artificial immune network technology. Based on the clone and affinity mutation principals of biological immunity mechanism, containing memory cells, the AINFCM is capability of maintaining local optima solutions and exploring the global optima defined as minimum of the objective function.

AINFCM algorithm works as an artificial immune network according to the immune network theory, originally proposed by Jerne [9], hypothesized a novel viewpoint of lymphocyte activities, natural antibody (Ab) production, pre-immune repertoire selection, tolerance, self/non-self discrimination, memory and the evolution of the immune system. The immune cells can respond either positively or negatively to the recognition signal. A positive response would result in cell proliferation, activation and antibody secretion, while a negative response would lead to tolerance and suppression. The artificial immune network is an edge-weighted graph, not necessarily fully connected, composed of a set of nodes, called antibodies, and sets of node pairs called edges with an assigned number called weight, or connection strength.

AINFCM algorithm explores optimal membership degrees matrix and centers of clusters with FCM minimum objective function shown in equation (1), under the constraint shown in equation (2), where $u_{ij} \in [0,1]$, indicates the membership of data vector x_i assigned to cluster *ith*. c_j is the *jth* center of C and d_{ij} is the Euclidean distance between data vector x_i and center c_j. Parameter $m \in [1, \infty)$ is so-called fuzzifier, which is used to control how much clusters are allowed to overlap.

$$J(U,c_1,...,c_c)=\sum_{i=1}^{c}J_i=\sum_{i=1}^{c}\sum_{j}^{n}u_{ij}^m d_{ij}^2 \quad (1)$$

$$\sum_{i=1}^{c}u_{ij}=1,\forall j=1,...,n \quad (2)$$

2.1 Problem Description

1) Antigen is unlabeled data. $Ag_{N\times P}\subset\Re^P$. N is the number of data, P is features' dimension for each data point.
2) Antibody represents a possible solution of clusters. $Ab_{M\times CP}\subset\Re^{CP}$, $CP=P*ClusterNumber$, M is the number of antibodies .
It is feasible to encode either center matrix C or membership function matrix U. The relationship between C and U can be denoted as:

$$c_i=\frac{\sum_{j=1}^{n}u_{ij}^m x_j}{\sum_{j=1}^{n}u_{ij}^m},\quad u_{ij}=\frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}}. \quad (3)$$

In this paper, we encoded C into antibodies of immune network accounting for calculation quantity.
3) Fitness evaluates a set of given centroids, which is inversely proportional to objective function *J*. The smaller FCM objective function *J* is, the better output centroids *C* are. *f* can be calculated as:

$$f=\frac{1}{J(Ab_i,Ag)+1},\quad f\in[0,1]. \quad (4)$$

4) Similarity of Ab_j to Ab_k describes the recognition level within antibodies, where $D(Ab_j,Ab_k)$ is the Euclidean distance described as:

$$Similarity(Ab_j,Ab_k)=D(Ab_j,Ab_k). \quad (5)$$

5) Offspring antibodies are identical copies of their parent cell. Selected best offspring antibodies will further suffer a somatic mutation so that they become variations of their parent. Clone parameter Nc and mutation rate β are proportional to the fitness of Ab, which determine the next generation of antibodies. The next generation Ab^* can be calculated as:

$$Ab^*=Ab+\beta N(0,1) \quad (6)$$

, Where $N(0,1)$ is a Gaussian random variable of zero mean and standard deviation $\sigma=1$.
6) Stop criterion can be set as a fixed iteration number of steps under the constraint that the difference of best fitness value between $(n+1)^{th}$ generation and n^{th} generation is small than ε:

$$\left\|f^{(n+1)}-f^{(n)}\right\|<\varepsilon. \quad (7)$$

2.2 Process of AINFCM

Step1: Initialize Immune network: Set antigens *Ag*; create a random population of M antibodies *Ab* and memory cells the parameters of fuzzy clustering.
Step2: While stopping criterion is not met do,
Step2.1: Calculate the fitness values of network cells and normalize the vector of fitness;
Step 2.2: Select best *n* antibodies to clone and mutate proportionally to their fitness values;
Step 2.3: Calculate the fitness of new generations;
Step 2.4: Reselect highest fitness cells with ς% from new generation;
Step 2.5: Suffer network suppression by removing cells with lower fitness value and higher similarity value to those cells with higher fitness value;
Step 2.6: Dynamically renew *d*% network cells.
Step3: Decode the optimal clusters and membership function.

3. PARALLELIZING AINFCM

After introduced the serial version of AINFCM, we turn our attention to parallelize this algorithm. The goal of the parallelization of the AINFCM algorithm is to discover the optimal cluster centers and to study how to improve efficiency. This ability will, in theory, allow us to apply AINFCM to problem sets of a larger scale without sacrificing some of the appealing features of the algorithm, as well as the implication of such work for the study of the role of diversity and distributions in artificial immune system.

From the process of AINFCM we can see that the most complex computation in the AINFCM algorithm are the fitness calculations in equation (1) and (4), which are performed in each iteration. From time cost analysis of serial AINFCM, it showed that calculation fitness values of antibodies and new generations after clone and mutation cost more than 90% run time. Therefore, steps 2.1 and 2.3 of the serial AINFCM algorithm should be parallelized. Our initial approach to paralleling is sketched in Fig.1 and described by the following sequence:
Step1: (Root Processor) Initialize immune network.
Step2: (Root Processor) Splits *Ab* antibodies among the available *p* processors and broadcast them so that each processor received M/p antibodies, where *M* is the number of antibodies, and *p* is the number of processors.
Step3: (All Processors) Calculate fitness values of antibodies.
Step4: (Root Processor) Gather fitness values and select best *n* antibodies to clone and mutate proportionally to their fitness values.
Step5: (Root Processor) Splits new generated antibodies *C* among the available *p* processors.
Step6: (All Processors) Calculate the fitness of portion of C antibodies.
Step7: (Root Processor) Reselect highest fitness cells with ς% from new generation. Suffer network suppression.
Step8: (Root Processor) Dynamically renew *d*% network cells. If the stop criterion is meet, stops, otherwise return to step 2.
Step9: (Root Processor) Decode the optimal clusters and membership function.
There are significant differences between our approach and the approaches to both parallel CLONALG and parallel AIRS.

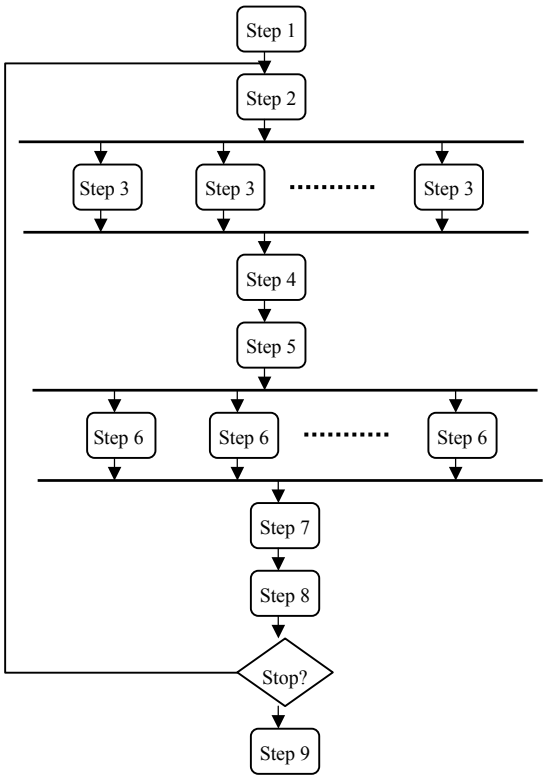


Fig. 1. The parallel AINFCM fuzzy clustering

For the parallelization of CLONALG, the training patterns are simply divided by the number of processes involved in the parallel job. Each process then evolves one memory cell respectively. For the parallelization of AIRS, it partitions the training data and allowing multiple copies of AIRS to run on these fractions of the data in essence creates some separate memory cell pools. There are no communications between processes in the evolution running time of parallel CLONALG and AIRS, so they can get considerable higher speedup rate. Unfortunately, parallel AINFCM employs some degree of interaction between processes for iterations, which leads to significant difference in behavior from the parallel version of CLONALG and AIRS. So, how to communicate with processes affects the overall performance of the algorithm.

4. RESULTS AND DISCUSSION

In the world of parallel computing the Message Passing Interface (MPI) is the de facto standard for implementing programs on multiple processors. MPI defines C and Fortran language functions for doing point-to-point communication in a parallel program. MatlabMPI [10] is set of Matlab scripts that implement a subset of MPI and allow any Matlab program to be run on a parallel computer. The key innovation of MatlabMPI is that it implements the widely used MPI "look and feel" on top of standard Matlab file i/o, resulting in a "pure" Matlab implementation. Thus, MatlabMPI will run on any combination of computers that Matlab supports.

The experiments presented in this paper were developed using Matlab6.5 and MatlabMPI 1.2. All of these were taken an average over 10 cross-validated runs and tested the parallel version on an increasing number of processors. There are a couple of observations to be made from this initial set of experiments. Foremost, for our current purposes, there is a gain in overall runtime of the algorithm by parallelizing it, and this speedup is achieved without any loss in clustering in terms of objective function.

4.1 Speed-up Analysis

The speed up evaluation of the Paralleled AINFCM algorithm was made in two test steps. In the first one, the objective was to observe the algorithm behavior increasing the number of records. In the second test the objective was to observe the behavior of the algorithm when increasing the number of attributes.

The datasets example was generated with the *nDexample* function located in fuzzy clustering toolbox for Matlab [11], which generated the random multi-dimensional data.

First of all, datasets of different number of records were used. The datasets had 1,000, 10,000, 50,000, 250,000 and 1000,000 records. A fixed number of attributes and a fixed number of generations were used. The datasets had 2 attributes and ran 20 generations. The speed-up results are shown in Table 1.

Table 1. Speed-up results for datasets with different number of records

Number of records	Number of Processors				
	1	2	4	6	8
1,000	1	1.42	2.62	1.93	1.85
10,000	1	1.53	3.07	4.96	4.82
50,000	1	1.61	3.20	5.08	5.46
250,000	1	1.67	3.53	5.20	6.61

The algorithm's speed up when processing the less number of records was clearly worse than the others. Its highest speed up value was 2.62 when using 4 processors for 1,000 records. The algorithm showed higher speed up values for a larger number of processors when processing datasets with larger number of records. When processing a small number of records, communications are too costly compared to the advantage of parallelizing the calculations of fitness values to antibodies, so the benefits of parallelism do not happen.

As showed in Table 1, using 6 processors, speed up values of more than 5.0 were achieved for databases with more than 50,000MB, which is a good gain for the overall parallel process. The speed up of the parallel program against the correspondent sequential one has an efficiency factor that gets as much closer to 0.9 as the database records increase, when considering a growing number of processors.

Table 2. Speed-up results for datasets with different number of attributes

Number of attributes	Number of Processors				
	1	2	4	6	8
2	1	1.35	1.6	1.55	1.43
4	1	1.42	1.78	2.64	2.50
8	1	1.57	2.37	3.77	3.82
16	1	1.72	3.11	4.09	4.57

To investigate the effect of the number of attributes in the parallel process, four datasets were used: 10,000 records with 2 attributes, 10,000 records with 4 attributes, 10,000 records with 8 attributes, and 10,000 records with 16 attributes. All processing were executed for 1, 2, 4, 6, and 8 processors. Results are presented in Table 2.

Table 2 illustrated that the number of variables have also affected the overall performance of the algorithm. The processing of datasets with small number of variables gets smaller speed up values. The processing of datasets with

larger number of variables results in better speed up values when using a larger number of processors.

4.2 Parameter Setting of Immune Network

To apply the AINFCM to any problem, a number of parameters for immune network have to be defined. The main parameters for the AINFCM were:

n: Number of selected n best antibodies;

N: Clone parameter, max clone number of selected antibody;

ts : Suppression threshold of immune network cells.

In this section, using two-dimensional dataset with 1,250 records generated by *nDexample* function, we intend to discuss and analyze how sensitive the parallel AINFCM is to these predefined parameters. In particular, it will be studied the influence of the parameters in the convergence speed and number of memory cells in the AINFCM network.

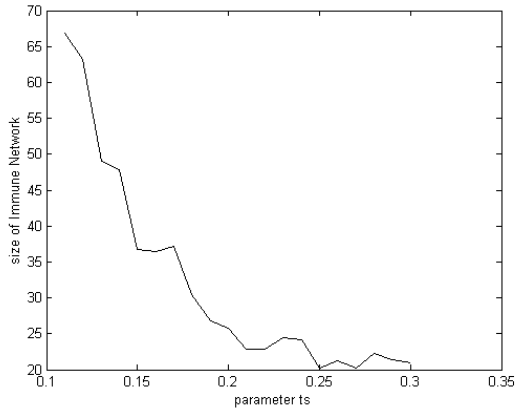


Fig. 2. Relationship between ts and network size.

Fig.2 showed relationship between suppression threshold ts and network size. Parameter ts affected the deletion of similar network memory cells with lower fitness, controls the final network size and the run time of algorithm. Too large value of ts leads to small network size and fast mature, while too small value results in larger immune network and increase run time.

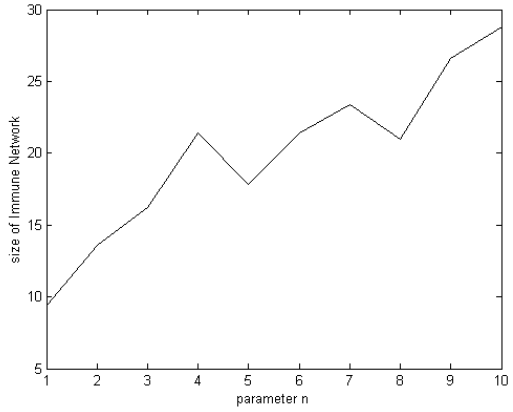


Fig. 3. Relationship between n and network size

In the each iteration, n best Ab with the highest fitness will be selected to undergo mutation. The clone rate and mutation rate are proportional to the fitness of each selected antibody. Fig.3 demonstrates introduced that larger value of n will improve the convergence speed of parallel AINFCM.

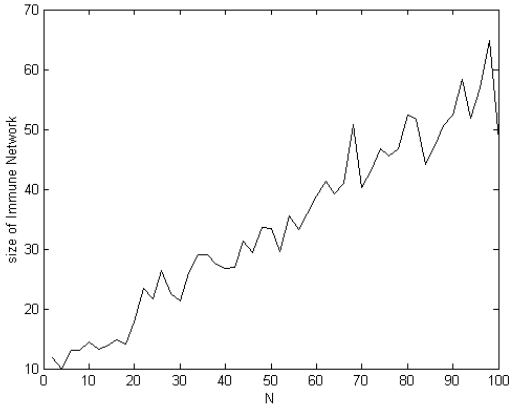


Fig. 4. Relationship between N and network size.

Fig.4 illustrates that the mutate parameter N is critical to control the iteration times of AINFCM final network size and is responsible for the convergence of the network. The amount of time to convergence increased as the value of *N* increases; however, the number of generations it took to reach this convergence seems to decrease as *N* increases.

5. CONCLUSIONS

This paper has shown that parallelization techniques can improve efficiency of Artificial Immune Systems. It was observed that when the number of records in the database grows, the processing time grows as well, proportionally to the increase of records. From experiments with AINFCM, we were able to achieve a great deal of reduction in processing time through some basic parallelization. Experiments indicated that processing with bigger files improved the speedup rate. It could be explained that the computational time tends to be greater than communications time involved in the process.

As for parallelization, Matlab MPI would be a good tool to run in the Matlab environment. However, its key innovation of implementation MPI communication on top of standard Matlab file i/o also decrease program speedup when parallelized among many processors.

With regards to future research we are focus on several topics. We plan to use program language C and MPI to implement the parallel AINFCM, to test among much more datasets and to expand the use of domain knowledge.

6. REFERENCES

[1] F. Klawonn, F. Hoppner, ”What is Fuzzy about Fuzzy Clustering? Understanding and Improving the Concept of the Fuzzifier”, In: MR Berthold:Advances in Intelligent Data Analysis, Springer,Berlin,2003,pp.254-264.

[2] A. k. JAIN, M.N. MURTY, “Data Clustering: A Review”, ACM Computing Surveys, Vol.31, No.3, Sept.1999.

[3] D. Castro, V. L. N. Zuben, “AiNet: An Artificial Immune Network for Data Analysis”, Idea Group Publishing, USA, 2001, pp.231-259.

[4] D. Castro, J. Timmis, “An Artificial Immune Network for Multimodal Function Optimization”, Proceedings of IEEE Congress on Evolutionary Computation (CEC’02), Hawaii, 2002, pp.699–674.

[5] A. S.Perelsen, G. F. Oster,: Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-Nonself Discrimination. J. Theor, Biol. 81(1979), pp.645–670.

[6] A. Watkins, X.T. Bi, A. Phadke," Parallelizing an

immune-inspired algorithm for efficient pattern recognition", Intelligent Engineering Systems through Artificial Neural Networks: Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Complex Systems and Artificial Life. Volume 13,New York, 2003, pp. 225–230.

[7] A. WATKINS, J. Timmis, " Exploiting Parallelism Inherent in AIRS, an Artificial Immune Classifier", ICARIS 2004: 427-438

[8] L. Liu, W.B. Xu, “Fuzzy Data Clustering Using Artificial Immune Network”, ICIC 2006, LNAI 4114,2006,pp.195–205.

[9] N. K. Jerne, “Towards A Network Theory of the Immune System”, Ann. Immuno, 125C, 1974, pp.373–389.

[10] J. Kepner,” Parallel Programming with MatlabMPI”, Lincoln Laboratory, Massachusetts Institute of Technology, http://www.ll.mit.edu/MatlabMPI/.

[11] B. Balazs, A. Janos et al, Fuzzy Clustering and Data Analysis Toolbox For Use with Matlab, http://www.fmt.vein.hu/softcomp/.

A parallel Multi-objective Genetic Algorithm On Cluster Computer*

Lianshuan Shi
Computer Department, Tianjin University of Technology and Education, Tianjin 300222, China
Email: shilianshuan@263.net
Hui Liu
Department of vocational Education, Tianjin University of Technology and Education, Tianjin 300222, China
Email: mylemonhat@163.com

ABSTRACT

A parallel genetic algorithm to solve multiobjective optimization with discrete design variables on distributed shared memory is introduced. Real-value coding is used and one of PCs group is used to initialize group. The selection procedure as well as the crossover procedure is finished by different PC. Non-dominated Sorting Genetic Algorithm is used to compute the fitness of chromosome. A few examples formed by randomly are used to test the efficiency of the algorithm. The CPU time and computing efficiencies in the condition of different PCs number and different population size are given.

Key words: Genetic Algorithm, Distributed share memory, parallel computing; Tread Marks

1. INTRODUCTION

Multi-objective optimization problem widely exists in practice. One way to obtain the optimal solutions of a multi-objective optimization is that the multi-objective optimization problem is converted into a single objective optimization problem firstly, and then a few mature algorithms are used to solve the single objective optimization problem which solutions are taken as optimum solution of the multi-objective optimization problem. Another way is to solve the Pareto efficient solutions of a multi-objective optimization problem. Fonseca and Fleming [1] have proposed a scheme in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. This blocked fitness assignment is likely to produce a large selection pressure that might produce premature convergence. The Non-dominated Sorting Genetic Algorithm was proposed by Srinivas and Deb [2] this algorithm is based on several layers of classification of the individuals. Before the selection is performed, the population is ranked on the basis of non-domination: all non-dominated individuals are classified into one category and they are shared with their dummy fitness values. Then this group of classified individuals is ignored and another nondominated individual is considered. The process continues until all individuals in the population are classified. In this algorithm, Non-dominated Sorting Genetic Algorithm is used to evaluate individuals.

* This paper was supported by the Education Committee Foundation of Tianjin (20022104).

Genetic algorithm is a search algorithm based on mechanics of natural selection and natural genetics [3, 4]. It uses probabilistic transition rules to guide itself toward an optimum solution. Compared to other search algorithms, GA has an excellent parallelism. Many parallel GAs have been developed since the traditional GA was proposed by Holland in 1975[5]. Genetic algorithms are easily parallelized algorithms. There are two kinds of possible parallelism—data parallelism and control parallelism [6]. Data parallelism involves the execution of the same procedure on multiple large data subsets at the same time. In contrast, control parallelism involves the concurrent execution of multiple different procedures.

For all their advantages shared memory multiprocessor computers have the disadvantage that they must be carefully designed and assembled and are therefore expensive and uncommon. Distributed memory multiprocessors can be assembled from ordinary computers on a high-speed network. Software interfaces like PVM and MPI make it possible to send and receive messages along the network from one processor to another and thus to consider a set of engineering workstations to be a distributed memory multiprocessor computer. For this reason distributed memory multiprocessing has become popular despite its complexity. TreadMarks is software that allows shared memory concurrent programming on a distributed memory multiprocessor computer or on a network of workstations. When shared memory is accessed on one processor, TreadMarks determines whether the data is present at that processor and if necessary transmits the data to that processor without programmer intervention. When shared memory is modified on one processor, TreadMarks ensures that other processors will be notified of the change so that they will not use obsolete data values. This notification is both immediate nor global immediate notification would be required too often and global notification would usually provide processors with information that was not relevant to the majority of them. Instead this notification takes place between two processors when they are synchronized Programs free of data races produce the same results whether notification is immediate or delayed until synchronization and the accumulation of many such notifications into a single message at synchronization time greatly reduces the overhead of inter processor communications.

2. DEFINITION OF THE EFFICIENT SOLUTION OF MULTI-OBJECTIVE PROGRAM

Let us consider the following multi-objective minimization problem: Minimize $z = F(X) = (f_1(X), f_2(X), \cdots, f(X))$

Subject to $X \in D$. Where, z is the objective vector, $f_i(X)$ is the $i-th$ objective to be minimized, X is the decision vector, and D is the feasible region in the decision space.

Suppose $X^* \in D$, if there does not exist $X \in D$, such that $f(X) \le f(X^*)$ （or $f(X) < f(X^*)$）, then X^* is called an efficient solution (or a weak efficient solution) of the multi-objective optimization problem. The set of all efficient solutions are noted R_{pa}, the set of all weak efficient solutions are noted R_{wp}. Efficient solution is also called Pareto optimal solution. Although there are many kinds of efficient solutions, but in practice, Pareto efficient solution and weak efficient solution are most commonly used.

3. PARALLEL MULTIOBJECTIVE GENETIC ALGORITHM ON SHARED MEMORY

The initialization of group is finished by one of PCs. In the process of initialization of population, firstly, the shared memory is open; every chromosome in population is stored in the share memory.

3.1 Coding

Coding in GA is the form in which chromosomes and genes are expressed. There are mainly two types of coding: binary and real. Here, real coding is used. Each optimization parameter is coded into a gene as a real number. The corresponding genes for all parameters form a chromosome, which describes each individual.

3.2 Evaluation of fitness

Fitness function is a designed function that measures the goodness of a solution. It should be designed in the way that better solutions will have a higher fitness function value than worse solutions. The fitness function plays a major role in the selection process. Nondominated Sorting Genetic Algorithm is used to evaluate the chromosome. At first, each solution of the population is classified in terms of dominance by other solutions. As usual, a nondominated solution does not allow the existence of any other better for all the objectives. In such a case, any improvement on one of the objectives must worsen at least one of the remaining ones. Then, through a ranking process, the solution fitness is determined according to the relative solution dominance degree. Basically the Pareto front, formed by the nondominated solutions, receives rank 1, and from the remaining population, the next nondominated solutions receive rank 2 and so on, until the entire population is ranked. Finally, the virtual fitness of each chromosome is obtained by inverting the chromosome rank level and the chromosomes with same rank have same fitness, that ensure those chromosomes with same ranks have same copy probability. Each PC computes a part of the fitness of chromosomes.

Algorithm as follow：
Begin
Rank←1;
For (i=start; i<=end; i++){
If there does not exist chromosome in the group, such that every objective of this chromosome is less than the corresponding objective of ith *chromosome, then the fitness of* ith *chromosome is*：
$f_i = M / rank;$
$rank++;$
}
End

3.3 Selection

A standard roulette wheel procedure is used to select crossover chromosomes. The selection operator chooses 100 pairs of chromosomes to be combined by the crossover. As for the crossover, a customized genetic operator is created to effectively combine the contents of a pair of parent chromosomes. For the kth chromosome with fitness f_k, its selection probability is computed by formula

$$p_k = f_k / \sum_{j=1}^{popZize} f_j .$$

The roulette wheel is made by these probability values. Then spin the wheel $popSize$ times, each times a new chromosome is selected and put into population.

3.4 Crossover

After being selected, crossover takes place. Actually, crossover operator plays a major role in GA, so defining a proper crossover operator is highly needed in order to achieve a better performance of GA. During this stage, Firstly, pair of mating individuals is selected at random. Their genetic material is exchanged in order to create the population of the next generation. The single point crossover is selected here.

3.5 Mutation

Mutation is another important genetic operator that randomly changes a gene of a chromosome. Mutation operator alters one or more gene in a chromosome. Mutation operator aims to achieve some stochastic variability of GA in order to get a quicker convergence. The probability of applying the mutation operator is usually set to be small. Here is 0.03.

4. EXPERIENCES AND RESULT

In this example, the number of population was 10000，the maximum number of generation was 100，crossover probability was 1，mutation probability is 0.3. The example is carried out on the cluster called Godzilla. The cluster consists of 32 PCs running Linux2.4, which are connected by a N-way100 Mbps Ethernet switch. Each of the PCs has 350 MHz processor and 192 Mbytes of memory.

Example 1： Consider a multi-objective program with 7 objective values and 20 design variables，

$$\min_{X \in D} f(X) =$$

$$(f_1(X), f_2(X), f_3(X), f_4(X), f_5(X), f_6(X), f_7(X))^T$$

where， $f_i(X) = \sum_{j=1}^{n} a_{ij} x_j = a_i[] \bullet X^T$,

$n = 20$， $X = (x_1, x_2, \cdots, x_{20})$,

$a_i[] = (a_{i1}, a_{i2}, \cdots, a_{in})$,

$a_1[] = (1, 2, 3, 6,4,11,2, 9, 1, 8, 5,1,2, 3, 5, 4,3,16,7, 3)$

$a_2[] = (5,1,2, 3, 5, 4,3,16,7, 3,1, 2,3,6,4,11,2, 9,1,18)$

$a_3[] = (8,1, 2, 3, 6,4,11,2,1, 8, 5,1,2,3, 5, 4,3,16,7,3)$

$a_4[] = (9, 5,1,2, 3, 5, 4,3,7, 3,1, 2, 3, 6,4,11,2, 9,1, 8)$

$a_5[] = (6,1, 2, 3, 6,4,2, 9,1,11, 8, 5,1,2, 3, 5, 4,3,16,7)$

$a_6[] = (16, 5,1,2, 3, 5,3,16,7, 3,1, 2,23, 6,4,11,2, 9,1, 8)$

$a_7[] = (4,8,1, 2, 3, 6,4,11,2,1, 5,1,12,3,35, 4,3,16,7, 3)$.

The discrete set is $D = (0,1,2,3,4,5,6,7,8,9)$.

Suppose number of PCs is p in the parallel system, T_s is the running time according to one PCs, T_p is running time according to p PCs, then the speedup ratio S_p is defined as $S_p = T_s / T_p$. The Table 1 shows the speedup ratio of computation and Table 2. gives computation time under different number of population.

Table 1. Speedup ratio of computation

Number of PCs	CPU time	Efficiency	Speedup ratio
1	1168.75		
2	766.98	1.52	0.76
4	460.82	2.54	0.64
6	394.58	2.96	0.49
8	323.38	3.62	0.45
Parameters	Number of population=10000 Maximum number of generations=50 Probability of Crossover=1 Probability of Mutation=0		

5. CONCLUSION

A Distributed Shared Memory system can give programmers the illusion of shared memory on the top of message passing distribute system, which facilitates the task of parallel programming in distributed systems [7]. Programmer need not consider how to deal with data communication between PCs. TreadMarks software can determines the data in a PC whether is translate to another PC. Because programming is easy and performing efficiency is higher, programmer can concentrate main attention on algorithm design. By the example, we can know, the algorithm can solve multi-objective combinatorial optimization problem in a higher efficiency. But when the number of PCs is increased, the efficiency is lower. By comparing running time between different number PCs, we found difference is very small when number of population is between 3000 to 4000. When the number of population is more than 6000, the computing time increase obviously.

Table 2. Computing time under different population

	Running time				
Population size / Number Of PCS	3000	4000	6000	8000	10000
1	83.14	83.31	367.5	714.4	1168.8
2	64.13	63.98	253.1	461.6	767.0
4	48.67	48.03	279.7	296.9	460.8
6	48.95	47.85	217.0	289.7	394.6
8	53.36	53.13	194.7	240.5	323.4

A view-Oriented parallel programming style for parallel programming on cluster computers is given by Z. Huang [8]. The advantage of this programming style is that that it offers the performance potential for underlying distributes shared memory system to optimize consistency maintenance. A significant performance gain of the programs based on the View-Oriented parallel programming style can be got. In the future, we will use VOPP to solve parallel multi-objective program.

ACKNOWLEDGEMENT

This paper is finished when author was a visiting scholar in Otago University, New Zealand. Thank Doctor Zhiyi Huang and his help.

REFERENCES

[1] C.M. Fonseca, P.J. Fleming, "An Overview of Evolutionary algorithms in Multi-objective Optimization", Evolutionary Computation., 1995，3(1)：1-16.

[2] N. Srinivas, K. Deb, "Multi-objective Optimization Using Nondominated Sorting in Genetic Algorithm", Evolutionary Computation. 1994，2(3)：221-248.

[3] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison Wesley, 1989

[4] J. Horn, N. Nafpliotis, et al, "A niched Pareto Genetic Algorithm for Multi-objective Optimization", Proc. Of 1st IEEE Conf. On Evolutionary Computation. 1994，82-87.

[5] J.H. Holland, Adaptive of Natural and Artificial Systems. Ann Arbor: The University of Michigan Press, 1975

[6] A.A. Freitas, Datamining and knowledge discovery with evolutionary algorithm, Spinger, Berlin, 264,2002.

[7] Àdora E. Calaor, Augusto Y. Hermosilla, et al, "Parallel Hybrid Adventures with simulated annealing and Genetic Algorithms", Proceedings of the international symposium on Parallel Architectures, Algorithms and Networks. 2002,1-6.

[8] Z. Huang, M. Purvist, et al, View-Oriented Parallel Programming on Cluster Computers, Proceeding of ICPP05, 2005.

Fast Sequential and Parallel Algorithms for Computing Scalar Multiplication in Elliptic Curve Cryptosystems*

Cheng Zhong, Runhua Shi, Lina Ge
School of Computer and Electronics and Information, Guangxi University,
Nanning, Guangxi, 530004, China
Email: chzhong@gxu.edu.cn , hfsrh@sina.com , gelina100@sina.com
Zhi Li
Guangxi Science and Technology Information Network Center
Nanning, Guangxi 530012, China
Email: lz@gxsti.net

ABSTRACT

The scalar multiplication is a key operation in the elliptic curve cryptosystem. The two fast sequential algorithms for computing scalar multiplication are presented and implemented by applying the dual partitioning principle and representing the key as a vector respectively; and a new parallel computing scalar multiplication algorithm is proposed and implemented by the divide-conquer principle and the balance tree technique. The experiment results show that the presented sequential algorithms are efficient and the parallel algorithm can obtain a good speedup.

Keywords: Elliptic Curve Cryptosystem, Scalar Multiplication, Sequential and Parallel Algorithms, Cluster Computing System.

1. INTRODUCTION

Miller [1] and Koblitz [2] proposed the elliptic curve cryptography (ECC) independently. The elliptic curve cryptography has been widely applied to the dada privacy and the network security communication. The foundation of the ECC security is the computational intractability of the elliptic curve discrete logarithm problem [3]. The basic operation in ECC systems is the computation of kP, where k is a large integer and P is an elliptic curve point. This operation is called scalar multiplication or point multiplication, which dominates the execution time of the elliptic curve cryptographic schemes.

With an arbitrary point or a fixed point P, some computing scalar multiplication methods exist [4,5]. For example, when P is an arbitrary point, there are some efficient methods such as the binary method, the binary-NAF method [6, 7] and all the kinds of window methods [8]. When P is known in advance, there are some efficient methods such as the fixed-base comb method [9] and the fixed-base window method [10]. Some parallel computing scalar multiplication methods have also been introduced in [11, 12, 13].

The remainder of this paper is organized as follows. In selection 2, the two fast sequential computing scalar multiplication algorithms are presented and evaluated. Selection 3 implements a parallel computing scalar multiplication algorithm. Selection 4 concludes the paper.

* This work is supported by Guangxi Science Foundation grant 0339008 and Guangxi Science and Technology Information Network Center

2. SEQUENTIAL ALGORITHMS FOR COMPUTING SCALAR MULTIPLICATION

2.1 An Improved Fixed-base Comb Algorithm

Suppose that the window size is w, P is a point over the elliptic curve. A binary expansion $(k_{l-1},\cdots,k_1,k_0)$ of an large integer k is partitioned into w groups denoted $k=b_{w-1}\|\cdots\|b_1\|b_0$, which group m is made of $(k_{md}, k_{md+1},\cdots, k_{(m+1)d-1})$ of length $d = l/w$,m=0~w-1. Based on the dual partitioning principle, the main idea of the improved fixed-base comb algorithm to compute scalar multiplication is that the each group for representing integer k is partitioned into h sub-groups which each sub-group includes d/h bits and these sub-groups are computed simultaneously in order to decrease the point doubling and addition operations in the main computation stage. We represent b_{mi}^{j} for the j-th bit in the i-th sub-group for group m, j=1~ d/h, i=0~h-1, m=0~w-1. The improved fixed-base comb algorithm for computing scalar multiplication is described as follows.

Algorithm1 the Improved Fixed-base Comb Algorithm for computing Scalar Multiplication

Input: w, k, P
Output: $Q = kP$
Begin
for j=1 to l/(w ×h) do // The precomputation stage
{ for i=0 to h-1 do
$S_i[j]=(b_{(w-1)(i-1)}^{j}, b_{(w-2)(i-1)}^{j},\cdots, b_{0(i-1)}^{j})$;
for i=0 to h-1 do
$S_i[j]P = b_{(w-1)(i-1)}^{j} 2^{((w-1)d+(i-1)d/h} P + b_{(w-2)(i-1)}^{j}$
$2^{((w-2)d+(i-1)d/h} P + \cdots + b_{0(i-1)}^{j} 2^{(0+(i-1)d/h)} P$;
}
Q = 0; // The main computation stage
for j =l/(w ×h) downto 1 do
for i =h-1 downto 0 do
{ Q=Q+ $S_i[j]P$;
Q=2Q};
Return Q
End.

2.2 An Fast Computing Scalar Multiplication Algorithm

Theorem 1 If k is a large integer and $0\leqslant k\leqslant n!$, k can be represented as $k=\sum_{i=1}^{n-1} k_i(i!)$, where $0\leqslant k_i\leqslant i$, and the expansion is called a factorial expansion of k.

Proof. Let $\omega = k$, $k_1 = \omega - 2\times\lfloor \omega/2 \rfloor$, $\omega_1 = \lfloor \omega/2 \rfloor$, $k_2 = \omega_1 - 3\times\lfloor \omega_1/3 \rfloor, \omega_2 = \lfloor \omega_1/3 \rfloor, \ldots, k_{n-1} = \omega_{n-2} - n \times \lfloor \omega_{n-2}/n \rfloor$, $\omega_{n-1} = \lfloor \omega_{n-2}/n \rfloor$. We have

$\sum_{i=1}^{n-1} k_i(i!) = (\omega - 2\lfloor \omega/2 \rfloor)1! + (\omega_1 - 3\lfloor \omega_1/3 \rfloor)2! + \cdots + (\omega_{n-3} - (n-1))\lfloor \omega_{n-3}/(n-1) \rfloor)(n-2)! + (\omega_{n-2} - n\lfloor \omega_{n-2}/n \rfloor)(n-1)! = \omega - n! \times \lfloor \omega_{n-2}/n \rfloor$, and

$\lfloor \omega_{n-2}/n \rfloor = \lfloor \lfloor \omega_{n-3} \rfloor/(n(n-1)) \rfloor$
$\le \lfloor \omega_{n-3}/(n(n-1)) \rfloor$
$\le \cdots \le \lfloor \omega/n! \rfloor$
$= 0$ ($\omega = k < n!$).

Therefore, $\sum_{i=1}^{n-1} k_i(i!) = \omega = k$ holds. □

From theorem 1, a positive integer k can be represented as $k = \sum_{i=1}^{n-1} k_i(i!)$, where $0 \leqslant k_i \leqslant i$. In fact, the cost of computing point addition is the same as the one of computing point subtraction on the elliptic curves. Thus, in the factor expansion, if k_i can be extended into a signed integer like the binary NAF representation, the computation of scalar multiplication can be speeded up.

Theorem 2 If k is a large integer and $0 \leqslant k \leqslant n!$, k can be represented as $k = \sum_{i=1}^{n-1} k_i(i!)$, where $|k_i| \le \lfloor (i+1)/2 \rfloor$, and the expansion is called a signed factorial expansion of k, denoted $SFE(k)$.

Proof. Given $(i+1)(i!) = (i+1)!$, it can be easily proved from theorem 1. □

Let us first describe in detail the procedure to compute $SFE(k)$ before presenting a new computing scalar multiplication algorithm:

Procedure 1 The Computation of $SFE(k)$
Input: n,k;
Output: coefficient k_i in a signed factorial expansion of k.

```
Begin
  ω = k ;
  k_1 = ω − 2·⌊ω/2⌋;
  ω_1 = ⌊ω/2⌋;
  c=0;
  s=0;
  for i=2 to n-1 do
    { s = ω_{i-1} − (i + 1) * ⌊ω_{i-1} /(i + 1)⌋;
      s=s+c;
      ω_i = ⌊ω_{i-1} /(i+1)⌋;
      if (s > ⌊(i+1)/2⌋) then
        { k_i =s-(i+1);
          c = 1;}
      else
        { k_i =s;
          c=0;}
    }
  output (k_1 , k_2 , …, k_{n-1});
End.
```

Clearly, computing the expansion of a large integer k spends $O(n)$ time.

Since a large integer k is represented as $k = \sum_{i=1}^{n-1} k_i(i!)$, where $|k_i| \leqslant \lfloor (i+1)/2 \rfloor$, scalar multiplication kP can be computed in the following way:

$$\begin{aligned} kP &= (\textstyle\sum_{i=1}^{n-1} k_i(i!))P \\ &= k_1P_1 + k_2P_2 + \cdots + k_{n-2}P_{n-2} + k_{n-1}P_{n-1} \\ &= \textstyle\sum_{d=1}^{h} d(\sum_{i:k_i=d} P_i) \\ &= \textstyle\sum_{d=1}^{h} d \cdot Q_d \\ &= Q_h + (Q_h + Q_{h-1}) + \cdots + (Q_h + \cdots + Q_1) \end{aligned}$$

where $Q_d = \sum_{i:k_i=d} P_i$, $P_i = i! \cdot P$, $i=1\sim n$, $h = \lfloor n/2 \rfloor$.

Based on the idea for representing a large integer k by a vector, we present a new sequential algorithm for computing scalar multiplication including a precomputation stage and a main computation stage. In the first stage, the elliptic points $P_1 \sim P_n$ are computed and stored in a table. In the second stage, the multiple-point is computed by using this table.

Algorithm 2 Computing Scalar Multiplication Algorithm with a Signed Factorial Expansion of k

Input: n,k,P;
Output: $Q=kP$.

```
Begin
  // The precomputation stage
  Q=P;
  for i=2 to n-1 do
    { Q=iQ;
      P_i=Q; }
  // The main computation stage
  Q=O;
  A=O;
  B=O;
  for d=h downto 1 do
    { for i=0 to n-1 do
        if  k_i=d then
          B=B+ P_i ;
        A=A+ B;
    }
  Q=A;
  return Q;
End.
```

2.3 The Algorithm Performance Evaluation

For a 160-bit integer, since $40! < 2^{160} < 41!$, let $n=41$. The estimation costs of sequential computing scalar multiplication in algorithm 1 and algorithm 2 are given in Table 1..

From Table 1., we can see that the number of the required doubling operations in algorithm 1 is smaller than that of the fixed-base comb algorithm; the required elliptic operations and the level of stored points in algorithm 2 are equivalent to the fixed-base window methods, which fit scalar multiplication when P is a fixed point, but algorithm 2 can be easily implemented in parallel. Therefore, we can design a parallel algorithm to speed up the computation of scalar multiplication by applying the MBB (Multiple-Base Binary) method [13].

Since $t=t_1+t_2$, where t is the required time of computing scalar multiplication, t_1 is the required time of computing the elliptic operations and t_2 is that of computing the expansion of a large integer k or computing the coefficient k_i of k, $i=1\sim n$-1, we can assume that if $t_2=0$, the total required time of computing scalar multiplication will be remarkably reduced. Based on this idea, the key is represented as a vector. For algorithm 2, in the implementation stage, a coefficient k_i ($i=1\sim n$-1) of the key vector (k_1, k_2,..,k_{n-2},k_{n-1}) is generated randomly so that $t_2=0$, and the speed of computing scalar multiplication will be improved remarkably.

Table 1. The estimation costs of sequential computing scalar multiplication algorithms (160-bit)

The methods	w	Stored Elliptic Points	Elliptic Operations (Av.)	
			Doubling	Addition
Fixed-base comb	4	30	19	36.5
Fixed-base window	4	40	0	50.5
Algorithm 1	4	54	10	36.5
Algorithm 2	-	40	0	54.99

2.4 The Experiment Results

The sequential computing scalar multiplication algorithms are implemented on a microcomputer system with Linux operating system by C++ programming. They uses elliptic curve over field of $F_{2^{163}}$ recommended by NIST: $y^2+xy=x^3+ax^2+b$,$a,b \in F_{2^{163}}$ and $b \neq 0$.

The experiment results are given in Table 2.. Table 2. shows that algorithm 1 is faster than the fixed-base comb method and algorithm 2 speeds averagely up 51.99% than the fixed-base window method with the equivalent level of stored points.

Table 2. The costs of sequential computing scalar multiplication algorithms (m=163)

The methods	w	Stored Elliptic points	Time (ms)
Fixed-base comb	4	16	9.049
Fixed-base window	4	41	5.946
Algorithm 1	4	29	4.763
Algorithm 2	-	40	3.912

3. PARALLEL ALGORITHM FOR COMPUTING SCALAR MULTIPLICATION

3.1 The Parallel Computing Algorithm

We assume that the parallel computing system has p processors. Based on the divide-conquer principle and the balance tree technique, the parallel computing scalar multiplication algorithm can be described as follows:

Algorithm3 Parallel Computing Scalar Multiplication

Begin

(1) Compute the SFE(k) by Procedure 1:

$k=\sum_{i=1}^{n-1} k_i(i!)$, *where* $|k_i| \leqslant \lfloor (i+1)/2 \rfloor$;

(2) for i=1 to p do in parallel

Compute $Q_i=\sum_{j=0}^{\lfloor (n-1)/p \rfloor} k_{p\times j+i} P_{p\times j+i}$ *by applying the MBB (Multiple-Base Binary) method [13];*

(3) //Compute $kP=\sum_{i=1}^{p} Q_i$ *in parallel.*

(3.1) for h=1 to $\log_2 p$ *do*

for i=1 to $p/2^h$ *do in parallel*

$Q_i \leftarrow Q_{2i-1} + Q_{2i}$;

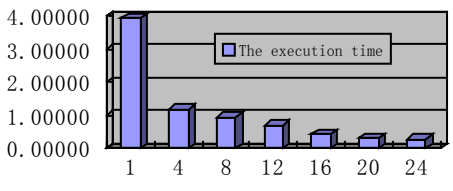
(3.2) $kP=Q_1$

End.

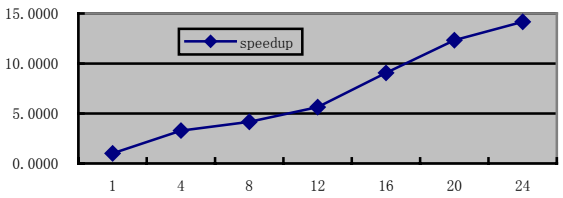
To compute Q_i in step (2) requires l_m point doublings and $\sum_{j=1}^{\lceil (n-1)/p \rceil} H(k_{p\times j+i})$ point additions, where $l_m=\max\{l_{p\times j+i} | 0 \leqslant j \leqslant \lceil (n-1)/p \rceil\}$, $l_{p\times j+i}$ is a bit length of binary representation of $k_{p\times j+i}$, and $H(k_{p\times j+i})$ is a binary Hamming weight of $k_{p\times j+i}$, i=1~p. For step (3), it requires $\lceil \log_2 p \rceil$ point additions to compute the final result. Therefore, the total elliptic operations are l_m point doublings and $\sum_{i=1}^{\lceil (n-1)/p \rceil} H(k_{p\times j+i}) + \lceil \log_2 p \rceil$ point additions.

3.2 The Experiment Results

With C-binding MPI programming [14] on a cluster computing system of personal computers, the presented parallel computing scalar multiplication algorithm is implemented and evaluated. The experiment results for the parallel execution time and speedup are described in Fig. 1.. Fig. 1. (*a*) gives the parallel execution time (m=163) and Fig. 1. (*b*) represents the corresponding speedup when the number of the computing nodes is 1, 4, 8, 12, 16, 20 and 24 respectively. Fig. 1.shows that the parallel computing scalar multiplication algorithm is feasible and gets a good speedup.



(*a*) The vertical coordinate represents the parallel execution time (millisecond), and the horizontal coordinate denotes the number of computing nodes.



(*b*) The vertical coordinate represents the speedup , the horizontal coordinate denotes the number of computing nodes.

Fig. 1. The execution time and speedup for parallel computing scalar multiplication algorithm

4. CONCLUSION

This paper presents an improved Fixed-base Comb algorithm for computing scalar multiplication by the dual partitioning principle and design a new sequential computing scalar multiplication algorithm by applying a strategy that a large integer is expanded into a signed factorial expansion and it is represented by a vector over the basis (1!, 2!, 3!,…,(n-1)!). Correspondingly, the scalar multiplication of a large integer can be obtained from that of a group of small signed integers, which can be computed in parallel. The experiment results show that the presented sequential computing scalar multiplication algorithms are efficient and the parallel computing scalar multiplication algorithm on the cluster computing system is feasible and obtains a good speedup.

5. References

[1] V. Miller, "Uses of elliptic curves in cryptography", Advances in Cryptology-CRYPTO'85, Lecture Notes in Computer Science, Berlin: Springer-Verlag, Vol.218, 1986, 417-426.

[2] N. Koblitz,"Elliptic curve cryptosystems", Mathematics of Computation, Vol.48, 1987, 203-209.

[3] D. Johnson, A. Menezes, et al, "The Elliptic Curve Digital Signature Algorithm", International Journal of Information Security, Vol.1, No.1, 2001, 36-63.

[4] M. Brown, D. Hankerson, et al, "Software implementation of the NIST elliptic curves over prime fields", Topics in Cryptology—CTRST 2001, Lecture Notes in Computer Science, Berlin: Springer-Verlag, Vol.2020, 2001, 250-265.

[5] D. Hankerson, J. L. Hernandez et al, "Software Implementation of Elliptic Curve Cryptography Over Binary Fields", Available at http://palms.ee. princeton. edu/fiskiran/repository/CHES/hankerson00software.pdf, 2001.

[6] F. Morain, J. Olivos, "Speeding up the computations on an elliptic curve using addition-subtraction chains", Informatique théorique et applications, Vol.24, 1990, 531 -544.

[7] J. Solinas, "Efficient arithmetic on Koblitz curves", Designs, Codes and Cryptography, Vol.19, 2000, 195 -249.

[8] A. J.Menezes, P. C. Oorschot, et al, Handbook of Applied Cryptography, London: CRC Press, 1996.

[9] E. F. Brickell, D. M. Gordon, et al, "Fast exponentiation with precomputation", Proceedings of EURO-CRYPT'92, Lecture Notes in Computer Science, Berlin: Springer-Verlag, Vol.658, 1993, 200-207.

[10] C. H. Lim, P. J. Lee, "More flexible exponentiation with precomputation", Advances in Cryptology-Crypto'94, Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1994, Vol.839, 95- 107.

[11] D. R. Stinson, "Some observations on parallel algorithms for fast exponentiation in GF(2^n)", SIAM Journal on Computing, Vol.19, 1990, 711-717.

[12] J.Von, Z. Gathen,"Efficient and optimal exponentiation in finite fields", Computational Complexity, Vol.1, 1991, 360-394.

[13] K. Kobayashi, H. Morita, et al, "Multiple Scalar-Multiplication Algorithm over Elliptic Curve", IEICE Transactions on Information and Systems, E84- D(2) , 2001, 271-276.

[14] G.L. Chen, H. An, et al, The Parallel Algorithms Practice, Beijing: Higher Education Press, 2004.

Research and Implementation on DSCP Classifier LFB of ForCES-based Router

Dengfeng Zhu, Weiming Wang, Jin Yu
College of Information and Electronic Engineering
Zhejiang Gongshang University
149 Jiaogong Road, Hangzhou, P.R.China, 310035
Email: dfzhu@pop.zjgsu.edu.cn wmwang@mail.hzic.edu.cn

ABSTRACT

One of the biggest challenges to the Internet is to provide QoS in a wide variety of applications. DSCP classifier is a key element for routers to implement Diffserv QoS. ForCES is a prospective architecture for next generation routers. A solution of DSCP classifier based on ForCES architecture is proposed. The framework and model of the ForCES-based router is introduced firstly. Then, the function of DSCP Classifier LFB is described, and the implementation of it in ForCES-based router is given in details. Finally, a test platform is implemented, and based on which the experiment result is fetched that has shown the feasibility of the DSCP Classifier LFB.

Key words: ForCES-based router, framework, model, QoS, DSCP Classifier LFB.

1. INTRODUCTION

With the rapid development of multi-media technology, there emerge more and more multi-media applications, such as IP phone, video conference, and remote education. Internet has evolved into a complex transmission network integrated with data, voice and other multi-media information. These different applications require different QoS, whereas, the internet only provides best-effort transfer services currently. It cannot provide QoS guarantee and effective network resource management. Routers which are the core of network need a revolution to adapt to this trend urgently. Traditional routers basically have two types: one is built on general processor which is flexible enough but has limited processing power; the other is built on ASIC (Application Specific Integrated Circuit), on the other side of the coin, which has high performance but limited programmability.

In conclusion, traditional routers can't provide high performance and flexibility simultaneously; also, it cannot provide QoS effectively. ForCES [1] (Forwarding and Control Element Separation) is a Working Group in IETF (Internet Engineering Task Force) routing area. ForCES proposed router architecture with the physical separation of the Forwarding Element (FE) and the Control Element (CE). Routers with this architecture meet the demands of the next generation networks very well, and such architecture is one trend of next generation routers.

2. FRAMEWORK AND MODEL OF FORCES-BASED ROUTER

Framework and model of ForCES-based router are defined in ForCES Requirement[2] (RFC 3654) and ForCES Framework[3] (RFC 3746). In essence, a ForCES-based router is a network element (NE). As the Fig.1 shows, it is made up of one primary CE, some redundant CEs and lots of FEs. The interface between CE and FE is defined by ForCES protocol. CE mainly processes ForCES protocol messages and manages various LFBs in FEs; FE mainly processes and transmits packet.

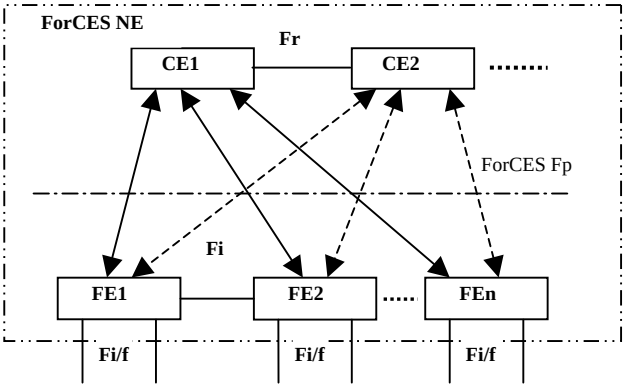


Fig. 1. Architecture of a ForCES router

FE architecture is defined in FE Model [4]. The diagram in Fig.2 shows the FE architecture. Resources inside an FE are expressed as kinds of LFBs which have different functions and interconnected to each other. LFBs may form different topologies that lead to different datapath for packets, which means that we can provide different services by arranging LFB into specific topologies. LFB topology and attributes are controlled by CE through ForCES protocol. Representative LFBs include classifier, scheduler, queue manager, forwarder and so on.

In ForCES FE Model, when a packet passes through a LFB, the LFB will process the packet according to the pre-defined behaviors. FE Model allows us to depict functions of FE by the blocks of LFB, so we can build complex network functions to implement different network services. An abstract LFB showed in Fig.2 have inputs, outputs and attributes which can be queried and processed through Fp reference point and ForCES protocol termination point.

* Project sponsored by National 863 Project (2005AA121310) , NSF China (60573116), and Zhejiang Key Sci&Tech Project (2005C21013)

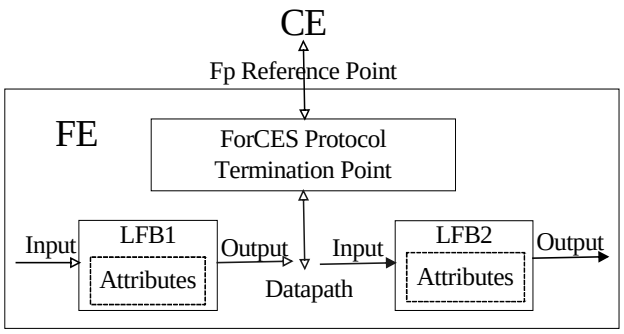


Fig. 2. FE architecture

Now, the research and implementation on DSCP Classifier LFB of ForCES-based router are presented in detail. Firstly, the paper depicts the function of DSCP Classifier LFB. Secondly, the implementations are presented. It presents from two sides: one is related work on ME [5] (Micro Engine) of FE; the other is related work on XScale of FE. Thirdly, the testing platform and the testing result are presented.

3. PARAMETERS OF DSCP CLASSIFIER LFB

DSCP Classifier LFB is a building block of DiffServ service. The main purpose is to classify packets according to DSCP field of IP header. The working flow is as follows: first, process packets according to classifier rule and then put the classifier result into the metadata for subsequent LFB to process, which mainly are queue management LFBs. In this LFB, the packet with an identical DSCP value represents a data flow. We can define corresponding classifier rule to provide different QoS for each flow which has different DSCP value.

3.1 Interfaces of DSCP Classifier LFB

All the DSCP Classifier LFB rules are stored at the SRAM table. The table is indexed with <input port, DSCP>pair. A single entry contains a complete classification result. This ensures that for each packet only one lookup in the table is required.

1) Input Variables: Table 1 lists the variables input to the micro block.

Table 1. Input Variables

.Variable	Size	Type[1]	Description
dl_input_port	16 bits	D	Logical number of an incoming interface, on which a packet was received.
dl_next_block	8 bits	D	Either hard-coded in a former block, or computed in run-time.
ip_header_in	2 bytes	X	An input xbuffer array with a cached IPv6 or IPv4 header. Only the first two bytes are used.

1. D = Dispatch loop variable, X = Xbuf array

2) Output Variables: Table 2 lists the variables output to the micro block.

Table 2. Output Variables

Variable	Size	Type[2]	Description
dl_next_block	8 bits	D	Identifier of a microblock that should continue with packet processing. The binding should be hard-coded in system.h as: • CLASSIFIER_DSCP_METER • CLASSIFIER_DSCP_MARK • CLASSIFIER_DSCP_IP4_FWD • CLASSIFIER_DSCP_IP6_FWD • CLASSIFIER_DSCP_INV_IP The output block is configurable on a per-rule basis.
dl_flow_id	32 bits	D	Identifier of a traffic flow to which the packet belongs. Subsequent blocks can use this value to select a policy instance.
dl_class_id	16 bits	D	Identifier of a QoS aggregate that share the same ordering constraint——that is, should be placed in the same queue. It is a relative queue number within an output port.
dl_color_id	2 bits	D	Packet drop precedence level: green, yellow or red color.

2. D = Dispatch loop variable

3.2 DSCP CLASSIFIER LFB RULE

Fig .3 shows the layout of DSCP CLASSIFIER LFB rule.

Fig.3 Layout of DSCP CLASSIFIER LFB rule

flow_id						
class_id	res.	IPC	S	TC	R	color_id

The DSCP Classifier LFB only reads the rules, while the core component (CC) updates them.

4. IMPLEMENTATION

4.1 Related work on Micro Engine of FE

Fig.4 shows the position of DCSP Classifier LFB in our system:

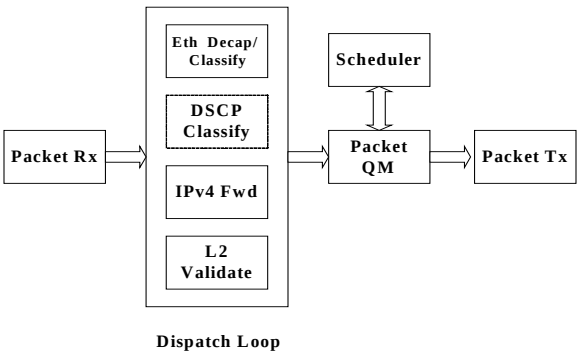


Fig.4 The position of DCSP Classifier LFB in system

DSCP Classifier LFB as a function module with IPv4 Fwd LFB runs in the dispatch loop. In order to CE can dynamically control the addition and deletion of DSCP Classifier LFB, we have a shared control block between Ethernet decapsulation microblock and DSCP Classifier LFB. This control block is used to dynamically set dl_next_block value for Ethernet decapsulation micro block.

The process of DSCP Classifier LFB dealing with a packet is showed as follows:

1. Examining whether the parameter dl_next_block is DSCP Classifier, if not, we do nothing;

2. Extracting the field of DSCP value and gaining the arrived interface of packet, then , according to the index of rule, send read command to gain the result of query table;

3. After classifying the packet, DSCP Classifier LFB will put the classification result into metadata for subsequent QoS model to use. For example, we put the different flows into different queues and use appropriate schedule algorithm to provide differ sever for different flows.

4. Finally, check whether the input interface is enabled. If not, according to the IP version of packet, set the dl_next_block as CLASSIFIER_ DSCP_IP4_ DEFAULT _ OUTPUT or CLASSIFIER_DSCP_IP6_DEFAULT _ OUTPUT. For example, the IPv4 packet is processed by 6_TUPLE EXACT MATCH CLASSIFIER LFB and the IPv6 packet is processed by IPv6 6_TUPLE EXACT MATCH CLASSIFIER LFB. Otherwise, check whether the input interface is set with a TC (Traffic Conditioning) flag. If the TC value is 1, set the dl_next_block as Meter; if the TC value is 0 and the Remark value is 1, set the dl_next_block as Marker; if the TC and Remark flags are all 0, set the dl_next_block as IPv4 or IPv6 forwarder according to the version of IP packet.

4.2 Related work on XScale of FE

There is a special CC (Core Component) on XScale to configure DSCP Classifier LFB .The main configuration commands are showed as follows:

1) Enable the DSCP Classifier LFB
EnableDSCPClassifierMB ()
2) Disable the DSCP Classifier LFB
DisableDSCPClassifierMB ()
3) Set the DSCP rule on the enabled interface
SetDscpRule ("<input_port> <dscp> < flow_id > < class_id > < color_id > < TC flag > < Remark flag > < Statistics Flag >")

For example, SetDscpRule (“0 1 5 8 1 1 0 1”). It means that set the input interface as 1, the output flow_id as 5, the class_id as 8, the color_id as 1; the TC flag as 1(it means that the next LFB is TC Meter), the statistics flag as 1(it means that statistic the number of packet which the DSCP value is 1).

4) Read the statistic information of the special rule at the interface
GetDscpStats ("<port> <dscp>")
For example, GetDscpStats ("0 1"). It means that read the number of packet whose rule id is 1 and interface is 0.

5. TESTING ANALYSIS

In order to validate the work we have done, the testing platform is constructed as the Fig.5 showed.

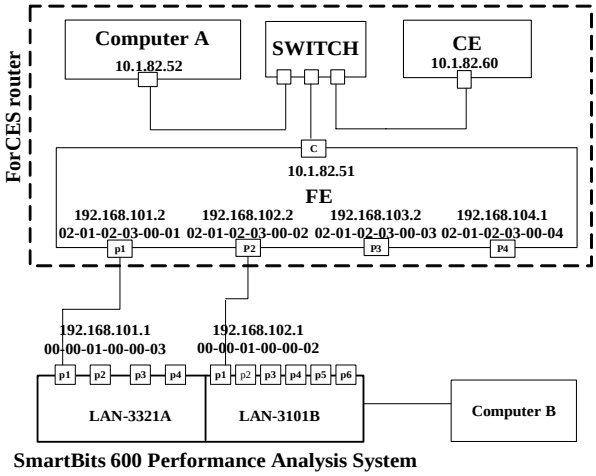


Fig.5 testing platform

As the Fig.5 shows, CE is a common PC, running Windows XP. It uses Microsoft Visual C++ 6.0 as the development tool. FE is a Intel IXDP2401 development board with four gigabits interface. It is used to receive and send network traffic. VxWorks running on IXDP2401 and Computer A with Tornado installed make up of the cross-development environment. CE, FE and Compute A are interconnected by a 10/100M Ethernet switch. SmartBits 600 Performance Analysis System has two boards. One is LAN3321A which have two gigabits interfaces (1-01 and 1-02); the other is LAN3101B which have six 100M interfaces. These two boards can receive and send network flows. In the test, we send packets from port1 of LAN3321A on SmartBits 600 Performance Analysis System to port1 on FE. Then, FE processes these packets and sends them back to port1 of LAN3101B on SmartBits 600. Computer B which installs the SmartBits analysis software is used to analysis network traffic.

5.1 Testing Approach

1) Set the LFBState of DSCP Classifier LFB as ON on CE. FE receive this configure message, then, call the function: EnableDSCPClassifierMB() on XScale to enable the DSCP Classifier LFB.
2) On CE, Configure two DSCP rules and send them to FE. As Fig.6 shows, we set the first rule as follows: input_port = 0，dscp =1，flow_id = 1，class_id =2，color_id = 0，TC flag, Remark flag and Statistics Flag are default value; set the second rule as follows: input_port = 0，dscp =2，flow_id = 2，class_id =8，color_id = 0，TC flag, Remark flag and Statistics Flag are default value. We can see the DSCP value of the first flow is 1 and the DSCP value of the second flow is 2, that is to say, flow 1 is priority to flow 2.

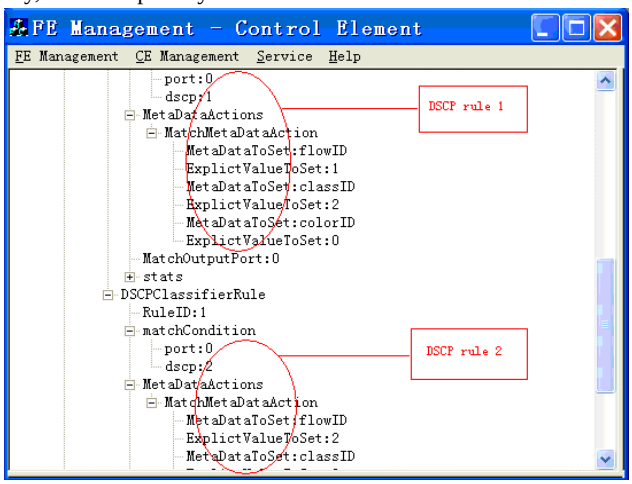


Fig.6 Two DSCP rules

3) Configure two DSCP flows (Flow1 and Flow2) on Smart Flow as the Fig.7shows.

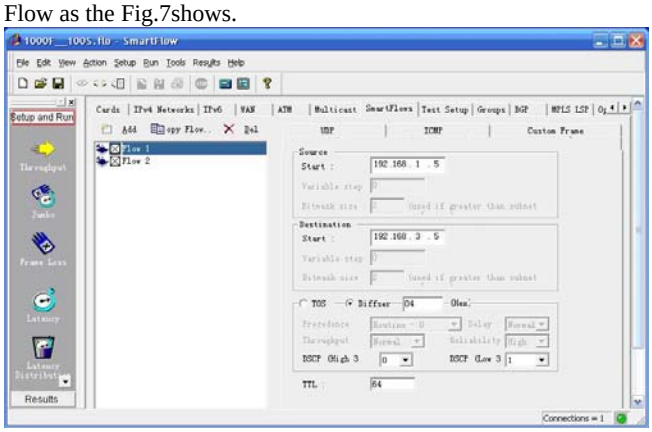


Fig.7 Configuration on Smart Flow

4) Smart Flow sends two flows which have different DSCP values to port1 of FE through port1 of LAN3321A. Then, LAN3321A transmits these packets to port2 on FE through DSCP Classifier LFB. Finally, FE sends these packets to port 1 of LAN3101B through port2 on FE.

5.2 Testing Result

In SmartFlow, the DSCP value of Flow1 (Group3) is 1 and the DSCP value of Flow2 (Group4) is 2. According to the DSCP Classifier rule, Flow1 is put into the high priority queue and the Flow2 is put into the low priority queue. The result of two flows' average delay is showed as Fig.8. When the traffic is low, the delay of two flows is the same. But, when the traffic approaching the limit of the output link between port 1 of FE and port1 of SmartBits, the delay of the low priority flow is much bigger than the high priority flow. Testing result proves that DSCP Classifier LFB of ForCES-based router implements the specified function of DiffServ model.

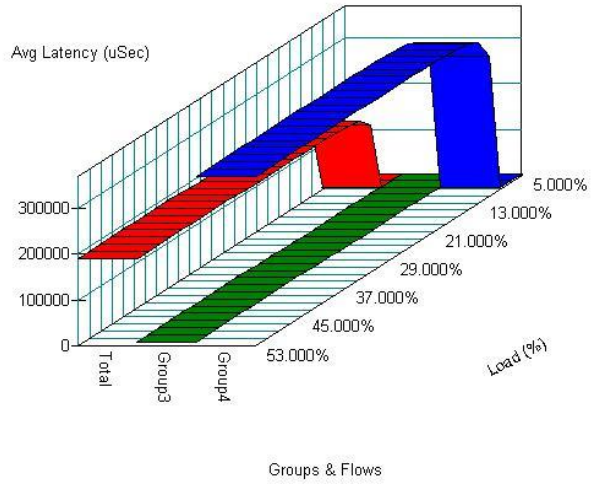


Fig.8 Comparison with the average of flow's delay

6. CONCLUSIONS

The implementation of DSCP Classifier LFB of ForCES-based router provides important foundation for standardizing the ForCES protocol and provides reference for researching ForCES protocol. Whereas, the implementation is abecedarian. There are lots of things we should to improve: 1) Implement some schedule algorithms (such as DER, WRR and WFQ) to perfect QoS model; 2) Due to the ForCES protocol has not been as RFC, there are lots of things to be standardized. On the one hand, we continue to participate in formulating the ForCES protocol. On the other hand, we try our best to improve our system according to the latest ForCES protocol.

7. REFERENCES

[1] A. Doria, R. Haas, et al, "ForCES Protocol Specification", March 23, 2006 http:// www.ietf.org/ internet drafts/ draft-ietf-forces- protocol-08.txt
[2] Khosravi, T. Anderson, "Requirements for Separation of IP Control and Forwarding", RFC 3654, November 2003. http://www.ietf.org/rfc/ rfc3654.txt?number=3654
[3] L.Yang, , Dantu, et al, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, April2004. http://www.faqs.org/rfcs/rfc3746.html
[4] L.Yang, , J. Halpern, et al, "ForCES Forwarding Element Model", Feb. 2005. http://www.ietf.org/internet-drafts/draft-ietf-forces-model-06.txt
[5] Intel, "Intel Internet Exchange Architecture Portability Framework Developer's Manual", 2004

Parallel Multipole Method To Solve A Special Eigen problem

Chunfeng Liu1,2, Aimin Yang1, Lamei Tong3, Shaohong Yan1, Yali He1
1, College of science, Hebei Polytechnic University, Tangshan, Hebei Province, 063000 China
2, Hebei University of Technology, P.R. China , Tianjin, 300130 China
3, College of Science, Yanshan University, Qinhuangdao, Hebei Province, 066004 China
Email: liucf403@163.com, aimin@heut.edu.cn

ABSTRACT

For a special type of matrix such as $A = D + \rho zz^t$, its eigenvalues are the roots of a function essentially of the form $\phi_{dc}(\lambda) = 1 + \sum_{i=1}^{n} \frac{z_i^2}{d_i - \lambda}$. Using multipole method to evaluate this function, the calculation of its eigenvalues was speeded up .In the research, the parallel multipole method on the base of multipole method is used and a better numerical result is received when the matrix is big, and the speed of this algorithm is nearly four times as much as the Newton iterative method.

Key words: Multipole Method; Parallel Multipole Method; eigenvalues; Symmetric Eigenproblem; FMM.

1. INTRODUCTION

FMM has two forms: uniform FMM works well if the particles are distributed evenly. Usually they will not be, and then we use Adaptive FMM. Greegard and Rokhlins suggest the Adaptive FMM, which is used to solve the N-body problem (please confer literatures [1,2]). Uniform FMM can be simply parallelized using simple domain decomposition (please confer literatures [3,4]). In this paper, we add parallel thoughts to the FMM, and use parallel Multipole method to solve the eigenproblem of the matrix: $A = D + \rho zz^t$, we receive a much better numerical effective.

Where D is an $n*n$ diagonal matrix, Z a vector, and ρ a scalar. The eigenvalues of this type of matrix are the roots of the function (please confer literature [5]):

$$\phi_{dc}(\lambda) = 1 + \rho \sum_{i=1}^{n} \frac{z_i^2}{d_i - \lambda} \qquad (1)$$

The d_i are the diagonal elements of D and Z_i are the elements of the vector Z. There is exactly one root in each interval $[d_i, d_{i+1}]$, except that if $\rho > 0$ there is a root in $[d_n, \infty]$ and none in $[-\infty, d_1]$, and vice versa if $\rho < 0$.

2. PARALLEL MULTIPOLE ALGORITHM

2.1 FMM

Multipole methods have been described in many other works, so we will only describe the simplest version of Multipole.

This is a 1-dimensional, non-adaptive, slow multipole method .We will call each (q_i, x_i) a particle, with strength q_i and location x_i. Suppose that all the particles are in the interval $[0,1]$, this is divided into a tree of subintervals .On the zero-th level there is the one interval $[0,1]$, and the intervals on the i-th level of the tree are generated by cutting each interval on the $i-1$-th level into two equal halves, which are called its children .The contributions in each subinterval I is approximated by a multipole function of the form (please confer literature[6]):

$$m(x) = \sum_{k=0}^{p} \frac{a_k}{(x-c)^{k+1}} \qquad (2)$$

Where c is the center of I, and the coefficients are determined by interpolating at $x = \infty$, which gives,

$$a_k = \sum_{z_i \in I} q_i (z_i - c)^k \qquad (3)$$

On the highest level of refinement, the multipole functions are constructed by directly summing the contribution of the particles in the interval, this takes pl operations .at coarser levels, the function coefficients are generated by fusing together the two multipoles which are the children of a subinterval. If a_{k1}, a_{k2}, b_k are the child and parent multipole coefficients, and c_1, c_2, c_p are the centers of the respective intervals, then,

$$b_l = \sum_{i=1}^{2} \sum_{k=0}^{l} a_{ki} (c_i - c_p)^{k-l} \binom{l}{k} \qquad (4)$$

To evaluate the function at a point x, the contribution of the particles, which are in the same subinterval as x are summed directly .The particles in the two neighboring intervals are also summed directly. The contribution of all other particles is approximated by a sum of different multipole functions on different levels of the tree. The choice of which multipoles to use in this sum is organized by the concept called a interaction list, the interaction list of an interval I is all intervals of the same size which are the children of the neighbors of It's parents, but are not neighbors of I. Suppose x is in I, first the multipoles of the intervals on I's interaction list are summed, then those on the interaction list of I's parents, and I's parents's parents, all the way to the top of the tree. The number of operations required to evaluate $h(x)$ at n points is $O(pn) + O(pn \log n)$. The derivative $h'(x)$ can also be rapidly computed using multipole methods. No new multipole coefficients need to be generated.

2.2 The data structure of the FMM

In FMM, we decompose the computing domain recursively to obtain the tree structure. In one dimension, we obtain two same intervals for each splitting, then eventually denoted by a quadric-tree. In FMM, the main data structure is tree.

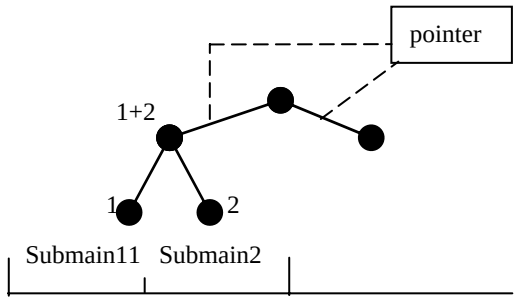
In our experiments, we use a 10-level full tree to decompose the whole domain into 512 basic subintervals. In the tree, each node denotes a subinterval. It has the following information:
Center of the subinterval, coefficients of the FMM, the farther pointer (Fig. 2.1.).

2.3 The paralleling of the FMM

The computation of the FMM mostly centralizes at two steps, and also, the parallel work we use in our experiment is mainly in these two steps.

1) The decomposition of the domain and the computation of the FMM coefficients

During the construction of the quadric-tree, we recursively decompose the solution domain, the quadric-tree can be divided into several separate subtrees. And there will be not conflict among the initialization of the separate subtrees. Because we do not use the FMM coefficients of the top two levels of the tree, so we take each node in the third level as a root, and divide the tree into four subtrees and parallelly initialize them. This is a kind of big grain paralleling (Fig. 2.2.).



Centre of submain 1+2

Fig. 2.1.

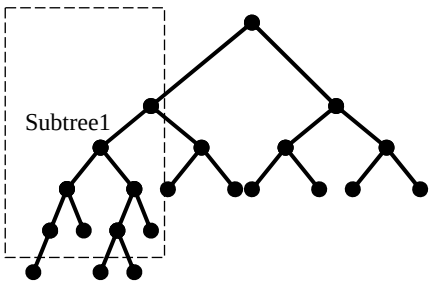


Fig. 2.2.

2) Using the already known FMM coefficients to solve

The process of the algorithm is also the use of the FMM tree that already constructed .It is the use of the coefficients and the reading of the relationship among the sub domains. During this period, we do not write new data to the tree, so there will be not conflict between writing and reading. So each eigenvalue solving process is separate, and we assign the n-1 eigenvalues evenly to several parallel process; that is to say, we use large grain paralleling. (Here we use four parallel processes.)

3. NUMERICAL RESULTS

We constructed random matrices of the form $A = D + \rho zz^t$, where d_i were uniformly distributed in the interval $[0,1]$, and the z_i^2 uniformly distributed in the interval $[0.01,1.01]$, and with $\rho = 1$ .We used Newton method to find the $n-1$ interior roots of the secular function $\phi_{dc}(\lambda) = 1 + \rho \sum_{i=1}^{n} \frac{z_i^2}{d_i - \lambda}$. This computation was performed three times, once using Newton iterative method, and once using the multipole method, and once using the parallel multipole method .The CPU time for these computations for s range of different matrix sizes is shown in the following Fig. 3.1.

4. CONCLUSIONS

We can see that, when the matrix is big, the parallel multipole method is faster than the other two methods .We can also see from the figure that the curve of the multipole is close to line, this is to say that the most CPU time is spent in the FMM generation step and the direct summation step. We use large grain parallel in our experiment, and use double super thread parallel machine, so the communication cost is very small, the acceleration ratio is close to 1.

At least in the symmetric problems, the parallel multipole method can speed up the computation of the eigenvalues effectively, but in many actual applications, the eigenvalues of the matrix do not distribute evenly in the interval, may centralize somewhere, but very few somewhere. In this case, we use Adaptive FMM.

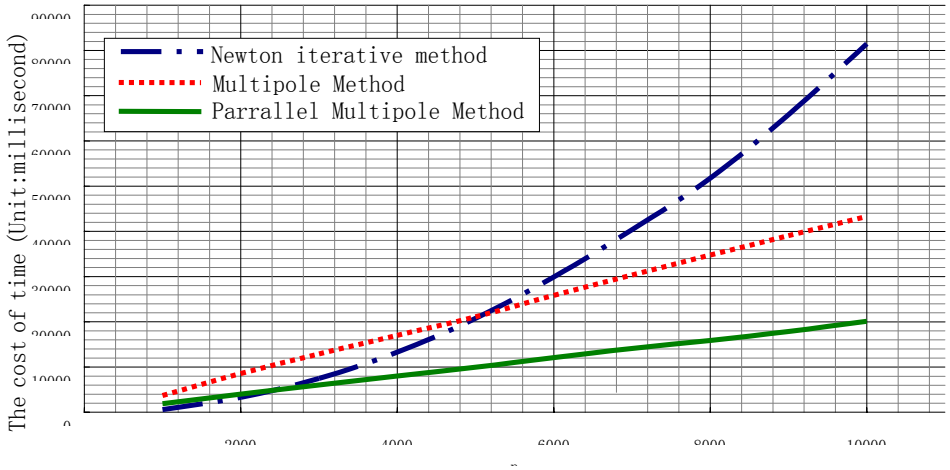


Fig. 3.1. The Comparison of Three Methods

5. ACKNOWLEDGEMENTS

Acknowledgements to my teacher Chen Yiming, Yu Chunxiao, and my firends. Because of their help, I can do the test of this paper successfully.

6. REFERENCES

[1] L.Greengard, V.Rokhlin, “A Fast Algorithm for Particle Simulations”. J. Comput. Phys , 73 (1987), pp.325-348.

[2] L.Greengard, W.Gropp, “Parrallel Processing for Scientific Computing” [J] SIAM, 1987, pp. 213-222.

[3] D.P.O ‘Leary,G.W.Stewart, “Computing the Eigenvalues and Eigenvectors of Symmetric Arrowhead Matrices”, J. Comp. Phys. 90, 1990，pp.497-505.

[4] G.X. Shen, “Multipole Mehod and Rolling Engineering ”. Beijing: Science Publishing, 2005.

[5] K. Yelick. “Applications of Parallel Computers Tree-Structured Codes for N-Body Simulations”, Berkeley, 2004.

[6] D.Y.Liu, G.X. Shen. Three Dimensional FMM BEM. Computing Mechanics, 2004.

A Parallel Particle Swarm Optimization Algorithm

Yan Ma, Jun Sun, Wenbo Xu
Institute of Information Technology, Southern Yangtze University, WuXi, 214122, China
Email:mayan1616@163.com

ABSTRACT

Particle swarm optimization algorithm is a newly proposed population -based algorithm. Although efficient in many optimization problems, it may encounter the problem of premature convergence and computational time consumption. In this paper, we attempt to introduce parallel mechanism into PSO and proposes PPSO (Parallel PSO) algorithm. We test the PPSO on four widely known benchmark functions and the experiment results show the efficiency and efficacy of PPSO.

Keywords: particle swarm optimization algorithm, parallel, parallel particle swarm optimization, function test.

1、INTRODUCTION

Numerical optimization has been widely used in engineering to solve a variety of NP-complete problems in areas such as structural optimization, neural network training, control system analysis and design, and layout and scheduling problems to name but only a few. In these and other engineering disciplines, two major obstacles limiting the solution efficiency are frequently encountered. First, large-scale problems are often computationally expensive, requiring significant resources in time and hardware to solve. Second, engineering optimization problems are often plagued by multiple local optima and numerical noise, requiring the use of global search methods such as population-based algorithms to deliver reliable results.

The Particle Swarm Optimization (PSO) algorithm is one of emerging global search methods [1, 2]. It is a kind of new swarm intelligence algorithm after Ant Algorithm and is particularly suited to continuous variable problems and has received increasing attention in the optimization community. It has been successfully applied to large-scale problems in several engineering disciplines and, as a population based approach, is readily parallelizable [3]. It also has fewer algorithm parameters than either GA or SA algorithms. Furthermore, Generic settings for these parameters work well on most problems [4].

This paper introduces the parallel mechanism to increases diversity of the swarm.

Through parallelization, PPSO not only improves the search ability of PSO, but also save computation time.

2、ClASSIC PSO ALGORITHM

Particle Swarm Optimization, originally proposed by Kennedy and Eberhart in 1995, is a population-based evolutionary computation technique, which differs from other evolution-motivated evolutionary computation in that it is motivated from the simulation of social behavior. In a PSO system, a particle corresponding to individual of the organism, which depicted by its position vector $\vec{x}$ and its velocity vector $\vec{v}$, the particles are manipulated according to the following equation:

$$v_i(t+1) = w_i(t) + \varphi_1(p_i - x_i(t)) + \varphi_2(p_g - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

$i = (1, 2, \cdots, M)$

Where x and v denotes the position and velocity of particle i among the population correspondingly, ${\varphi_1}^T$ and ${\varphi_2}^T$ are two random vectors within [0, 1].

In (1), vector p_i is the best position (the position giving the best fitness value) of the particle i and vector p_g is the position of the best particle among all the particles in the population. Parameter w is the inertia weight, which value is induced from 0.9 to 0.4 linearly in most cases.

3.PARALLEL PARTICLE SWARM ALGORITHM AND IMPLEMENT

The development in microprocessor technology and network technology has led to increased availability of low cost computational power through clusters of low to medium performance computers. To take advantage of this, communication layers such as MPI and PVM have been used to develop parallel optimization algorithms, the most popular being gradient-based, genetic (GA), and simulated annealing (SA) algorithms. In biomechanical optimizations of human movement, for example, parallelization has allowed previously intractable problems to be solved in a matter of hour. As a result of PSO algorithm is based on evolution algorithm of swarm, contains dormant parallel mechanism, so it can be paralleled and not limited to number of computers.

Parallel Particle Swarm Optimization algorithm put the high speed of Parallel computers with the parallel of particle swarm optimization, promoting the computational speed and quantities of PSO very distinctly.

3.1 Parallelization

Like many other evolutionary algorithms, the major problem confronts Particle Swarm Optimization Algorithm is premature convergence, which results in great performance loss and sub-optimal solutions. With PSOs the fast information flow between particles seems to be the reason for clustering of particles. Diversity declines rapidly, leaving the PSO algorithm with great difficulties of escaping local optima. Another problem with PSO in multi-modal optimization is computational cost. With the dimension of the optimization problem increasing, the population size must be enlarged to ensure the algorithm have a good performance, which, however, makes the algorithm computationally expensive.

To solving the aforementioned problems, we propose in this paper the Parallel Particle Swarm Optimization (PPSO).

The PPSO employ the co-evolution model, in which the global population (swarm) is partitioned into *q* subpopulations, where *q* is the number of PPEs (physical processing elements). The PPEs communicate periodically to exchange the *gbest* and the communication is a synchronous voting that the *gbest* of a subpopulation is broadcast to all the PPEs. The subpopulation stores the *gbest* received from other counterparts in its local memory and randomly selects a *gbest* at each iteration to and adjust its velocity and position according to the Eq. (1).

The period of communication for the PPEs is set to be T number of generations (iterations), which follows an exponentially decreasing sequence: initially $\lceil N_g/2 \rceil$, then $\lceil N_g/4 \rceil$, and so on, where N_g is the maximum number of generations. The rationale is that at the beginning of the search, the diversity of the global population is high. At such early stages, exploration is more important than exploitation; therefore, the PPEs should work on the local subpopulation independently for a longer period of time. When the search reaches the later stages, it is likely that the global population converges to a number of different *gbests*. Thus, exploitation of more promising position is needed to avoid unnecessary work on optimizing the local *gbest* positions.

With the above design considerations, the parallel Particle Swarm Optimization Algorithm is outlined below.

3.2 Parallel particle swarm optimization

The algorithm is as follows:

(1) Initialize a population (array) which including *m* particles, for the *i*th particle, it has random location X_i in the problem space and for the *d*th dimension of velocity V_i, v_{id} =*Rand2 ()* $*V_{\max}$, where *Rand2 ()* is in the range [-1, 1];

(2) Partition the population into q subpopulations with size equal to *M/q* and assign to the different PPEs, respectively. M is the size of the whole population;

(3) i =0;

(4) **Do**

(5) T=0;

(6) **Do**

(7) **For** each of the subpopulation do

(8). Evaluate the desired optimization fitness function for each particle;

(9). Compare the evaluated fitness value of each particle with its *gbest*. If current value is better than *gbest*, then set the current location as the *gbest* location. Furthermore, if current value is better than *gbest*, then reset *gbest* to the current index in particle array;

(10). Change the velocity and location of the particle according to the equations (1) and (2), respectively;

(11) **End for**

(12) **Until** ++T=$\lceil N_g / i \rceil$;

(13) Accept the *gbest* position from a remote PPE to replace the *gbest* of the subpopulation.

(14) $i = i \times 2$

(15) **Until** the total number of generations elapse equal to N_g

4.EXPERIMENTS AND RESULTS

4.1 Test function

To test the performance of parallel particle swarm algorithm, three benchmark functions are used here for comparison with SPSO, the three functions are： Rosenbrock Function, Rastrigrin Function, Griewank Function. These functions are all minimization problems with minimum value zero. For the purpose of comparison, Table 1. lists the initialization ranges and $V_{\max}$ and $X_{\max}$ values for all the functions, respectively. The fitness value is set as function value.

Table 1.

	Functions	Initial Range	$X_{\max}$	$V_{\max}$
F1	$\sum_{i=1}^{n} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[15,30]	-100	100
F2	$\frac{1}{4000}\sum_{i=1}^{n}(x_i - 100)^2 - \prod_{i=1}^{n}\cos(\frac{(x_i - 100)}{\sqrt{i}}) + 1$	[300,600]	-600	600
F3	$\sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i) + 10)$	[2.56,5.12]	-10	10

We had 50 trial runs for every instance and record mean best fitness values and run-time for 50 runs of each functions in Table 2. to Table 4.. In order to investigate the scalability, different population sizes M are used for each function with different dimensions. The population sizes are 20, 40 and 80. Generation is set as 1000, 1500 and 2000 generations corresponding to the dimensions 10, 20 and 30 for three functions.

Table 2. The mean fitness value and run-time for Rosenbrock Function

M	Dim	Gmax	SPSO		Two Processors		Four Processors	
			Mean best	Run time	Mean best	Run time	Mean best	Run time
40	10	1000	71.0239	0.408314	18.658	0.220722	9.4043	0.135542

	20	1500	179.291	1.25326	143.921	0.66117	142.438	0.369977
	30	2000	289.593	2.50315	410.585	1.53776	508.957	0.744269
80	10	1000	37.3747	0.788501	10.3677	0.427034	8.1202	0.232065
	20	1500	83.6931	2.49435	46.5886	1.27953	30.8569	0.677497
	30	2000	202.072	4.77487	74.7392	2.60106	91.8591	1.44484

Table 2. The mean fitness value and run-time for Griewank Function

M	Dim	Gmax	SPSO		Two Processors		Four Processors	
			Mean best	Run time	Mean best	Run time	Mean best	Run time
40	10	1000	0.08496	0.362269	0.073467	0.165142	0.0507879	0.11254
	20	1500	0.02719	0.871019	0.016211	0.474874	0.014069	0.26496
	30	2000	0.01267	1.67027	0.01045	0.893636	0.00818699	0.48275
80	10	1000	0.07484	0.65406	0.05814	0.32328	0.0406955	0.18364
	20	1500	0.02854	1.86063	0.0196612	0.957418	0.0168997	0.53245
	30	2000	0.01258	3.82191	0.0099579	1.9843	0.0106857	1.06545

Table 4. The mean fitness value and run-time for Rastrigin Function

M	Dim	Gmax	SPSO		Two Processors		Four Processors	
			Mean best	Run time	Mean best	Run time	Mean best	Run time
40	10	1000	3.5778	0.284396	2.12204	0.163872	2.56471	0.0981328
	20	1500	16.4337	1.010656	15.9206	0.447622	23.9366	0.248176
	30	2000	37.2796	1.694956	44.7115	0.824832	50.7817	0.522098
80	10	1000	2.5646	0.63688	1.15806	0.315888	1.0113	0.1779108
	20	1500	13.3826	1.957438	11.10809	0.918312	11.0227	0.565094
	30	2000	28.6293	3.9857	23.5322	2.06244	27.3771	1.03203

4.2 Analyze the test result

By compare the results, it is easy to see that PPSO have better result than SPSO when population size is larger than 20. When there are 20 particles in populations, the result of PPSO is not better than SPSO, for each processor's particle is less, not becomes a community. About the run-time, it is obvious to see that the time of PPSO was reduced than SPSO.

Compare the run-time of processor in Fig. 1. (Next figure express population sizes 80 are used for each function with 30 dimensions, generation is set as 2000, and the processor number is 1, 2 and 4.)

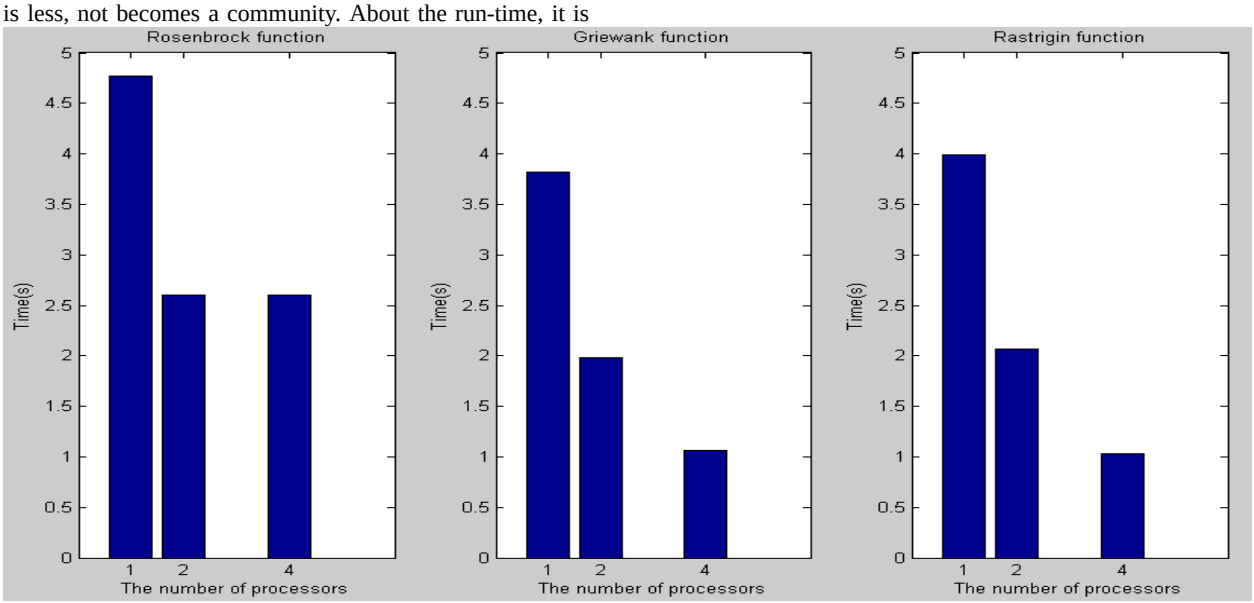


Fig. 1. The run-time for three functions

5、CONCLUSION

In this paper, parallel particle swarm optimization algorithm was introduced, at the base of parallel particle swarm optimization algorithm improved it, and the test result indicated that PPSO algorithm not only improved the searching ability of SPSO, but also the run-time was less than SPSO.

Future work will focus on approaches of solving large-scale problems or apply it to problems with the run-time demands strictly.

6,REFERENCES

[1] J. Kennedy, R. Eberhart, "Particle swarm optimization", Proc. IEEE Int. Conf. on Neural Networks, Perth, Australia, 1995, pp.1942-1948.

[2] R. C. Eberhart, Y.H. Shi, “Particle swarm optimization: Developments, applications, and resources”, Proceedings of the 2001 Congress on Evolutionary Computation, 2001, pp.81-86.
[3] J. Kennedy, “The Behavior of Particles”, Evol. Progr. VII (1998) pp. 581-587
[4] A.J. van Soest, L.J.R. Casius, “The merits of a parallel genetic algorithm in solving hard optimization problems”, Journal of Biomechanical Engineering 2003, 125:141-146.

Research and Realization of Complement Set Arithmetic for Multi-Valued Logic Function *

Jianlin Qiu, Bo Wang
School of Computer Science and Technology, Nantong University
Nantong, Jiangsu 226019, P.R. China
Email: qiu.jl@ntu.edu.cn

ABSTRACT

The optimization of multi-valued logic function is an extension of binary-valued logic function. Realization of multi-valued logic function is based on binary computer. In the complement set of logic synthesis, it is realized that non-completeness enumerate is transformed to completeness enumerate. In order to get the complement set of multi-valued logic function, the procedure is given out as bellow. At first, describes multi-valued variable and multi-valued logic function by binary vector. Secondly, transforms multi-valued cube into the Boolean expression. At last, calls recursion fission arithmetic to get the result. The time complexity of the arithmetic is $O(3^n)$, the space complexity is $O(2^{n+1})$.

Keywords: Arithmetic, Multi-valued Logic Function, Complement Set, Recursion fission.

1. Introduction

The complement set of logic function is frequently used in the logic function synthesis. This method is directly related with the optimization and efficiency of logic function arithmetic. The complement set is quite often called in logic synthesis [1,2,3], i.e., complement method is the best way to realize the optimization of logic function. For example, for a complete enumeration function, if $X^{ON} \cup X^{OFF}=U$, X^{ON} and X^{OFF} are relatively defined as complement set to each other. When one of the sets is given, then the other is out.

The recursion fission of the complement set is based on the sum-of-product expression [4]. It has three terminal conditions, which are derived from Shanno law. The complement set can be approached through the Cofactor expression [5]. The complement set in multi-valued logic [6,7] Function is based on the binary-valued logic function. Our primary task is to transform multi-valued function variables into binary function, to express the multi-valued cube in binary expression from the mini-item by calling recursion fission in complement set arithmetic.

Chapter Two discusses the multi-valued variables and expressions of multi-valued logic functions; Chapter Three discusses the transformation of a multi-valued cube; Chapter Four discusses the recursion fission in complement set arithmetic; Chapter Five analyses the experimental results and arithmetic process.

2. Multi-valued Variables and Expressions of Multi-valued Logic Function

In the expression of a multi-valued logic function, input n, output m

$$\text{MF: } \boldsymbol{S} \rightarrow \{0, 1, 2\}^m \text{S}= \overset{n}{\underset{i=1}{X}} S^i$$

Here, 0 means Negative (N) , 1 Positive (P) , 2 Irrelevant (I) .

Each S^i is an integer set $\{1,2, \ldots , S^i\}$, the input vector V in set (V1, V2, … , Vn) is an integral sequence. Vi, $1 \le i \le n$, is a multi-valued variable. The range is from 1 to S^i. In binary form, V_j^i （$1 \le j \le S_i$）, V^j means the jth of Variable V^i.

Definition 1: Multi-valued variable can be expressed with V_j^i in binary form. If Vi= j, then V_j^i =1; Otherwise, V_j^i =0.

Example 1: Suppose V^i, n=4, V^1、V^2 、$V^3$and V^4respectively respond to 2,5,4 and 3 , i.e. (S_1，S_2，S_3，S_4)=(2, 5, 4,3). If a vector V input gets (1,4,2, 3), then V^1 corresponds to 1 or in the first phase, V^2 to 4 or in the 4th phase, V^3 to 2 or in the second phase while V^4 to 3 or in the three phase. Here use V_j^i to express in binary form, get:

$$V = (v_1^1 \overline{v_2^1})(\overline{v_1^2} \overline{v_2^2} \overline{v_3^2} v_4^2 \overline{v_5^2})(\overline{v_1^3} v_2^3 \overline{v_3^3} \overline{v_4^3})(\overline{v_1^4} \overline{v_2^4} v_3^4) \ldots (1)$$

Or use a binary vector to express, get:

（10）（00010）（0100）（001） …(2)

Vector V can have only one definite variable, so V^i in binary has only one original variable, the rest are inverse variables. In Expression (2), in each parenthesis there is only 1, the rest are 0s.

Definition 2: Multi-valued variable V^i is expressed in a set of binary double-valued variables. Binary V_j^i has only one original variable, the rest are inverse variables, i.e., V_j^i expression is the mini-item expression of a multi-valued variable. Use 0 and 1 to define V_j^i in a binary vector expression.

Multi-valued variable might as well have cube form. For instance,

$$C = (\overline{V_1^1} V_2^1)(V_1^2 \overline{V_2^2} V_3^2 V_4^2 \overline{V_5^2})(V_1^3 \overline{V_2^3} V_3^3 \overline{V_4^3})(V_1^4 V_2^4 \overline{V_3^4}) \ldots (3)$$

In vector form is

(01)(10110) (1010) (110) … (4)

Notice that the major difference between Expression (3) and Expression (1) is that in each parenthesis there have many original variables in (3), while in (4) in its binary form, each parentheses allows many 1s.

Expression (3), (4) indicate that V^1=1, V^2=1, 3 or 4, V^3=1 or 3, V^4=1 or 2. Inside the parentheses are OR, between are AND, thus it is not a Boolean expression. While Expression (1) is not so, it is a Boolean expression itself. There needs a transformation into the Boolean expression before optimization.

3. Conversion of Cube

In the arithmetic of optimization, the input condition is given in cube set instead of Boolean expression. So it is a

* This work was supported in part by the Science Found of JiangSu Grants BK2001130 and the Science Found of JiangSu Education Department Grants 03KJB520103 and the Science Found of Nantong Grants K2006008.

must to transform them into Boolean expression before optimization.

Example 2: Study the mini-item set of multi-valued cube of binary system.

V^1_1	V^1_2	V^2_1	V^2_2	V^2_3
1	1	0	1	1

By Definition 2, the converted mini-item set is:

V^1_1	V^1_2	V^2_1	V^2_2	V^2_3
1	0	0	1	0
1	0	0	0	1
0	1	0	1	0
0	1	0	0	1

These mini-items are Boolean expressions. However, in a cube with n variables, as the V change from p_1 to p_n, the number of the mini-item set expanded is:

$$\prod_{i=1}^{n} 2^{p_i-1}$$

It is a very big number in many cases. So, is there a simple conversion?

Proposition 1: Suppose the function has n variables from V^1 to V^n, $V^i=(V^i_1, V^i_2, \ldots, V^i_{si})$, and they are in the sequence of 0, 1. Cube $C_m=V^1V^2\ldots V^n$, replace Sum $C_b=V^{1b}V^{2b}\ldots V^{nb}$ with $V^{ib}=(V^{ib}{}_1V^{ib}{}_2\ldots V^{ib}{}_{si})$

$$V^{ib}_j=\begin{cases}0 & V^i_j=0\\ 2 & V^i_j=1\end{cases}$$

0 stands for F(False), 1 for T(True), 2 for I(Irrelevant).

Replace 2 with 1, keep 0 constant.

For the binary multi-valued cube in Example 2, with regard to the mini-item, it can be expressed in the following Boolean expression:

V^1_1	V^1_2	V^2_1	V^2_2	V^2_3
2	2	0	2	2

In order to demonstrate its compatibility, we extend it in Boolean expression:

V^1_1	V^1_2	V^2_1	V^2_2	V^2_3	
0	0	0	0	0	
0	0	0	0	1	
0	0	0	1	0	
0	0	0	1	1	
0	1	0	0	0	
0	1	0	0	1	*
0	1	0	1	0	*
0	1	0	1	1	
1	0	0	0	0	
1	0	0	0	1	*
1	0	0	1	0	*
1	0	0	1	1	
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	

Please notice these "*"at the ends of Line 6, 7, 10 and 11, which means all the mini-items included in the cube. This is also true in Example 2, as other lines have no mini-item. So this replacement is compatible. The compatible replacement doesn't mean the replaced Boolean expression is completely equal to the original cube. But it is for the purpose of expansion of the basic irrelevant item.

4. Complementary Set by Recursion Fission

Definition 3: Establish the logic function f with n variables, $f=\sum A_i$, $1\le i\le m$, in matrix form is $\boldsymbol{M}(f)$. $\boldsymbol{M}(f)$ is the matrix of $m\times n$. (n for number of variables, m for number of sum-of-product). Each line corresponds to a sum, each column to a variable. For $A_i=a_1^i a_2^i \ldots a_n^i$, A_i is an accumulates of the item f, then $\boldsymbol{M}(f)$'s element is defined by the following expression.

$$M_{ij}=\begin{cases}0 & a_j^i=0 \text{ or } a_j^i=\overline{x}_j\\ 1 & a_j^i=1 \text{ or } a_j^i=x_j\\ 2 & a_j^i=\varnothing\end{cases}$$

Example 3: Establish Function $f=x_1\overline{x}_2x_4, + x_2x_3\overline{x}_4 + x_1\overline{x}_3x_4 + x_1x_3$, then its matrix form is

$$\boldsymbol{M}(f)=\begin{bmatrix}1&0&2&1\\2&1&1&0\\1&2&0&1\\1&2&1&2\end{bmatrix}.$$

Definition 4: In the matrix form of the logic function f, suppose Sum A to a sum set, $A=[a_1a_2\ldots a_n]$, Sum B to a sum set, $B=[b_1b_2\ldots b_n]$, then the cofactor of A to B is $(A)_B=C=[c_1c_2\ldots c_n]$, in which,

$$c_k=\begin{cases}\varnothing & a_k=\overline{b}_k \cap b_k\neq 2\\ 2 & a_k=b_k \cap b_k\neq 2\\ a_k & b_k=2 \cup a_k=2 \qquad 1\le k\le n\end{cases}$$

The logic function $f=\sum A_i$, its cofactor to B is the merge of respective items, i.e., $(f)_B=\sum (A_i)_B$

Example 4: Find the cofactors of $f=\overline{x}_2\overline{x}_4 + x_1x_3 + x_1x_2\overline{x}_3 + x_1x_2x_4$. The matrix form of the logic function f is

$$\boldsymbol{M}(f)=\begin{bmatrix}2&0&2&0\\1&2&1&2\\1&1&0&2\\1&1&2&1\end{bmatrix}$$

X_2 in vector form is [2 1 2 2], namely, b_1、b_3、b_4 is 2, b_2 is 1; So column 1, 3 and 4 of $\boldsymbol{M}(f)$ keep constant; Because Line 1 of column 2 of $\boldsymbol{M}(f)$ are 0s, therefore, the corresponding cofactors is $\varnothing$; Then there are only two lines of cofactors in matrix form as followed:

$$\boldsymbol{M}(f_{x_2})=\begin{bmatrix}1&2&1&2\\1&2&0&2\\1&2&2&1\end{bmatrix}$$

Theorem 1: The logic function f can be the sum-of-product of original variable and its cofactors plus the product of inverse variables and its cofactors expanded by any variable. (It is also known as Shanno theorem [3])

$$f = x_i f_{x_i} + \overline{x_i} f_{\overline{x_i}}$$

According to Theorem One, there are two important extended formulae.

Formula One: $\overline{f}=x_i\overline{f}_{x_i}+\overline{x_i}\overline{f}_{\overline{x_i}}$

Formula Two: $(\overline{f})_{x_i}=\overline{(f_{x_i})}$

By the cofactor Definition 4, f_{x_i} and $f_{\overline{x_i}}$, compared to f, the number of lines becomes less while the number of lines with number 2 is increasing.

To call the recursion of Theorem One, we will approach one of the following circumstances:

① When $f = \phi$, $\overline{f} = U$;

② when f has a line with 2 all in it, i.e., when f= U, then $\overline{f} = \phi$;

③ when f has one line only, then apply Morgan Theorem [3] to beg $\overline{f}$.

During the course of calling the recursion, also known as two-fork tree fission, the fission terminates when any one of the above circumstances appears, regression is needed immediately.

Complement Set Arithmetic by Recursion Fission

Step one: Set up a recursion two-branch tree with f as the root node to judge the termination of the nodes in those three circumstances. If conducted respectively, terminate the recursion and move to Step Three. If it is not any one of 3 above circumstances, then turn to the second step.

Step two: Select split variable x_i to make the column with x_i has the most 0 and 1. Let $\overline{x}_i$ to be left divaricator and get left sub-tree basis node. Let x_i to be right divaricator and get right sub-tree root node. The new node to be a root node, go back to Step one.

Step Three: Seek the corresponding $\overline{f}$ at the leaf node. multiply it with the variable from the father node's divaricator to generate an included item in complement set form, with the left and right leaf as the complement set of the father node. Turn to the fourth step.

Step four: Delete the left and right leaves when the complement set come into existence to make itself a superior leaf over father node till it becomes the root node, then follow the 5th step. Otherwise turn to the third step.

Step five: Output the complement set of function f of root node.

Example 5: Suppose $f = x_1 x_2 + x_2 \overline{x}_3 + \overline{x}_1 x_2 x_4 + \overline{x}_2 \overline{x}_4$, beg $\overline{f}$.

the matrix form expression is as follow:

$$\boldsymbol{M}(f) = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 2 & 1 & 0 & 2 \\ 0 & 1 & 2 & 1 \\ 2 & 0 & 2 & 0 \end{bmatrix}$$

The matrix form $\boldsymbol{M}(f)$ is expanded by root node . First select column 2 as the root nodes of left and right sub-tree in the fission of variables. This is because the 2th column includes the most 0 and 1. Then the matrix of $\boldsymbol{M}(f)$ to $\overline{x}_2$、x_2, beg the left and right sub- tree node are Matrix ① and ②respectively:

$$\begin{bmatrix} 2 & 2 & 2 & 0 \end{bmatrix} \quad \cdots\cdots ①$$

$$\begin{bmatrix} 1 & 2 & 2 & 2 \\ 2 & 2 & 0 & 2 \\ 0 & 2 & 2 & 1 \end{bmatrix} \quad \cdots\cdots ②$$

The node ①, which belongs to one of three circumstances, is called the leaf node.

The node ② corresponds to③、④ in matrix form during the course of begging cofactors of left and right sub- root node from $\overline{x}_1$ and x_1.

The corresponding right sub-tree root node in matrix forms of ② are ③、④ respectively.

$$\begin{bmatrix} 2 & 2 & 0 & 2 \\ 2 & 2 & 2 & 1 \end{bmatrix} \quad \cdots\cdots ③$$

$$\begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 0 & 2 \end{bmatrix} \quad \cdots\cdots ④$$

The node ④, which belongs to one of three circumstances, is called the leaf node.

The node ③ corresponds to⑤、⑥ in matrix form during the course of begging cofactors of left and right sub- root node from $\overline{x}_3$ and x_3.

$$\begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix} \quad \cdots\cdots ⑤$$

$$\begin{bmatrix} 2 & 2 & 2 & 1 \end{bmatrix} \quad \cdots\cdots ⑥$$

The nodes ⑤、⑥ which belong to one of three circumstances, are also called the leaf nodes. Till this is done with two-branch tree expanded, four leaf nodes are approached.

The leaf node ④ and ⑤ both includes a line with 2, then come to the corresponding node $\overline{f} = \phi$; Leaf node ①、⑥ contain only one line, beg each node respectively by Mogen theorem the results are respectively x_4、$\overline{x}_4$.

The collection of the data of the leaf nodes starts from the root node to non-empty leaf node path, the product of divaricator variable multiplied by non-empty leaf node. The result of this path brings out an included item, the begged merge of all the included items are the complement set of Function f. This collection is also done in recursion. Get:

$$\overline{f} = \overline{x}_2 x_4 + \overline{x}_1 x_2 x_3 \overline{x}_4$$

In matrix form $\overline{f}$ is expressed:

$$\boldsymbol{M}(\overline{f}) = \begin{bmatrix} 2 & 0 & 2 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

x3x4 / x1x2	00	01	11	10
00	1	0	0	1
01	1	1	1	0
11	1	1	1	1
10	1	0	0	1

Notice: This is only applicable to complete enumeration. For implicit enumeration, arithmetic is a better choice in improving it, because implicit enumeration has

$$U^{on} \cup U^{dc} \cup U^{off} = U$$

As $M^{on} = U^{on} \cup U^{dc}$, $M^{off} = U^{off}$,

Thus $M^{on} \cup M^{off} = U$, converted into complete enumeration, will combine the irrelevant items with the true value items, increase the number of root knots. Recursion fission is also used.

5. Analysis on the Arithmetic Applied

1. This method is comparatively complicated; the main task is to select fissionable variables, generation of two-branch trees and collection of data of leaf nodes, which are realized in recursion fission. The selection of fissionable variables will directly affect the depth of two-branch tree, a key factor in the process of arithmetic.

2. As it calls the recursion, the fission generates left and right sub-tree root nodes each time. The data from those nodes need backup. Meanwhile, the depth of two-branch tree determined the layers of recursion transfer nesting, thus the stack of system configuration have to be very big. This demands a large storage and high capacity system. The

space complexity is measured by nodes, the maximum is $O(2^{n+1})$.

3. Output $\boldsymbol{M}(\overline{f})$ is not the optimized result as redundancy left unfinished, i.e., complement $\boldsymbol{M}(\overline{f})$ is not the minimum coverage, the door leg price is not the best choice.
4. If measured with time complexity with regard to the selection of fission variables, generation of two-branch tree, data collection of leaf nodes, the time complexity is $O(3^n)$.
5. This approach is used in logic functions without any restriction, but it has a critical demand for the computer hardware. It is quite common in the complementary set of logic function.

6. Conclusion

The efficiency in optimization of logic functions is directly related to the complement set. How to eliminate the complexity of recursion fission in complementary set and the restrictions on the computer hardware is the major task in future research. The time consumed in this approach is mainly spent on the fission and regression collection while space needed is decided mainly by the depth of layers of two-branch tree in recursion transfer nesting. Thus the order in selection in fission variables has a certain effect on the time and space complexity in complementary set.

7. References

[1] J.S. Hong, R.G. Cain, et al, MINI: A Heuristic Approach for Logic Minimization.IBM Journal of Research and Development, 1974, 18(9): 443-458.
[2] R.B. Cutler, S. Muroga, Derivation of Minimal Sums for Completely Specified Functions.IEEE Transactions on Computers, 1987, 36(3) :277-292.
[3] M.Y. Liu, Theory of Computer-Aided Logic Design. Beijing: Science Press, 1985.
[4] B. Wang, Z.J. Guan, et al, Determining Essential During The Procedure of Generating Prime Implicants]. Proceedings of The 7th International Conference on Computer Aided Design and Computer Graphics, Kunming, China, August 22-24, 2001, 826-829.
[5] B. Wang, On the Identification of Essential Prime Implicants. Computer research and Development, 1995,32(12): 40-44.
[6] X.W. Wu, M. Pedram, Propagation Algorithm of Behavior Probability in Power Estimation Based on Multiple-Valued Logic [A]. IEEE: Proceedings of the 30th international Symposium on Multiple-Valued Logic, Portland, USA, May 2000, 453-459.
[7] T. Sasao, J.T. Butler, A Method to Represent Multiple-Output Switching Functions By Using Multi-Valued Decision Diagrams [A]. IEEE: Proceedings of the 26th International Symposium on Multiple-Valued Logic, Jantiago de Compostela, Spain, May 29-31, 1996, 248-254.

On Bivariate Quadratic Hermite-Padé Approximation*

Chengde Zheng, Zhibin Li
Department of Mathematics, Dalian Jiaotong University
Dalian, Liaoning 116028, P.R.China
Email: zhengchengde@dl.cn, lizhibin@dl.cn

ABSTRACT

The existence and local behavior is analysed of the off-diagonal bivariate quadratic Padé approximation to a bivariate function which has a given power series expansion about the origin. It is shown that the off-diagonal bivariate quadratic Hermite-Padé form always defines a bivariate quadratic function and that this function is analytic in a neighborhood of the origin.

Keywords: Hermite-Padé approximation, Bivariate Padé approximation, Bivariate analytic function.

1. INTRODUCTION

The Padé approximation theory has been widely used in problems of theoretical physics [1, 3], numerical analysis [6], and electrical engineering, especially in modal analysis model [2], order reduction of multi-variable systems [4].

This paper is concerned with the properties of the bivariate quadratic Hermite-Padé approximation. This approximation may be defined as follows (see, for example, Chisholm [5]).

Let $f(x,y)$ be a bivariate function, analytic in some neighborhood of the origin $(0,0)$, whose series expansion about the origin is known. Let $a_0(x,y), a_1(x,y), a_2(x,y)$ be bivariate polynomials,

$$a_k(x,y)=\sum_{i=0}^{m}\sum_{j=0}^{n}a_{ij}^{(k)}x^i y^j,\ k=0,1,2,$$

such that

$$E(x,y)=a_2(x,y)f^2(x,y)+a_1(x,y)f(x,y)+a_0(x,y)$$
$$=\sum_{(i,j)\in\mathbf{N}^2\setminus\mathbf{I}_{mn}}e_{ij}x^i y^j, \qquad (1)$$

where $\mathbf{I}_{00}=\{(0,0)\}$;

$$\mathbf{I}_{mn}=\{(i,j)\in\mathbf{N}^2: i+2j\le 3m+1, 0\le i\le 3m, 0\le j\le m\}$$
$$\bigcup\{(i,j)\in N^2: 2i+j\le 3n+1, 0\le j\le 3n, 0\le i\le m\},$$

for $(m,n)\in N^2,(m,n)\ne(0,0),0\le m\le n;$

or

$$\mathbf{I}_{mn}=\{(i,j)\in\mathbf{N}^2: 2i+j\le 3n+1, 0\le j\le 3n, 0\le i\le n\}$$
$$\bigcup\{(i,j)\in N^2: i+2j\le 3m+1, 0\le j\le n, 0\le i\le 3m\},$$

for $(m,n)\in N^2, m>n.$

Without loss of generality, we always assume that $m>n$ in this paper. Most of the results about $m>n$ are also true for $m<n.$

Note that such polynomials a_0,a_1,a_2 not all zero, must exist since (1) represents a homogenous system of $3(m+1)(n+1)-2$ linear equations in the $3(m+1)(n+1)$ unknown coefficients of the a_0,a_1,a_2. Then we set

$$a_2(x,y)u^2(x,y)+a_1(x,y)u(x,y)+a_0(x,y)=0 \qquad (2)$$

and attempt to solve this equation for $u(x,y)$ in such a way that $u(x,y)$ approximates $f(x,y)$.

In the well-known case of general Canterbury approximation [7], the same procedure is followed for

$$a_1(x,y)f(x,y)+a_0(x,y)=\sum_{(i,j)\in\mathbf{N}^2\setminus\mathbf{J}_{mn}}\tilde{e}_{ij}x^i y^j,$$

where

$$\tilde{e}_{i,2n+1-i}+\tilde{e}_{2m+1-i,i}=0,\ i=1,2,...,n; \qquad (3)$$

$$\mathbf{J}_{mn}=\{(i,j)\in\mathbf{N}^2: i+j\le 2m, 0\le i\le 2m, 0\le j\le n\}$$
$$\bigcup\{(i,j)\in N^2: i+j\le 2n, 0\le j\le 2n, 0\le i\le n\};$$

which gives $u(x,y)=-a_0(x,y)/a_1(x,y).$

Here,if $a_1(0,0)\ne 0$ (not a serious restriction), it then follows that

$$u(x,y)=f(x,y)+\sum_{(i,j)\in\mathbf{N}^2\setminus\mathbf{J}_{mn}}\tilde{e}_{ij}x^i y^j,$$

and Eq. (3) holds.

However, in the bivariate quadratic case it is not obvious that Eq. (2) yields even an analytic approximation to $f(x,y)$, still less that it defines a bivariate function $u(x,y)$ such that

$$u(x,y)=f(x,y)+\sum_{(i,j)\in\mathbf{N}^2\setminus\mathbf{I}_{mn}}e_{ij}x^i y^j.$$

The purpose of this paper is to show that an analogue of the Chisholm results is in fact true.

2. PRINCIPAL RESULTS

It will be assumed that Eq. (1) holds. Without loss of generality, we further assume that $a_0(x,y)$ doesn't vanish identically and

* Supported by the Science and Technology Development Foundation of Education Department of Liaoning Province (2004C060).

$$\sum_{l=0}^{2} |a_l(0,0)| > 0. \tag{4}$$

The following notation will be used:

(i) An approximation derived from (1) will be referred to as a quadratic $(m,n)-$approximation to $f(x,y).$

(ii) By $\sqrt{D(x,y)}$ we mean the principal square root of $D(x,y).$

(iii) Let

$$D(x,y) = a_1^2(x,y) - 4a_0(x,y)a_2(x,y).$$

If Eq. (2) holds, then

$$u(x,y) = \left(-a_1(x,y) \pm \sqrt{D(x,y)}\right) \Big/ \left(2a_2(x,y)\right)$$

and

$$\pm\sqrt{D(x,y)} = 2a_2(x,y)u(x,y) + a_1(x,y) = \frac{\partial}{\partial u}\left(\sum_{l=0}^{2} a_l(x,y)u^l(x,y)\right).$$

The problem divides itself into two cases, the case $D(0,0)=0$ and the case $D(0,0)\neq 0.$

2.1 The Case $D(0,0)=0$

This implies that $a_2(0,0)\neq 0$ (since if $a_2(0,0)=0$, then $a_1(0,0)=0,$ which with

$$\sum_{l=0}^{2} a_l(0,0)u^l(0,0) = 0$$

gives $a_0(0,0)=0;$ this contradicts the assumption Eq. (4).)

In this case we only treat one special case: $D(x,y)\equiv 0.$ Then from Eq. (1) we have

$$\left(2a_2(x,y)f(x,y)+a_1(x,y)\right)^2 = 4a_2(x,y)\sum_{(i,j)\in \mathbf{N}^2\setminus \mathbf{I}_{mn}} e_{ij}x^i y^j.$$

Through trivial calculation, finally we get

$$u(x,y) = -\frac{a_1(x,y)}{2a_2(x,y)} = f(x,y) + \sum_{(i,j)\in \mathbf{N}^2\setminus \mathbf{I}_{kl}} \overline{e}_{ij}x^i y^j,$$

and Eq. (2) holds, where

$$k = \max\{t\in\mathbf{N}: t\le m/2\}, l = \max\{t\in\mathbf{N}: t\le n/2\}.$$

2.2 The Case $D(0,0)\neq 0.$

Firstly we can easy prove the following result:

Theorem 1. If $D(0,0)\neq 0,$ then there exists a unique function $u(x,y),$ analytic in a neighborhood of the origin, satisfying Eq.(2) and $u(0,0)=f(0,0).$

With Theorem 1, we can prove

Theorem 2. If $D(0,0)\neq 0,$ then there exists a unique function $u(x,y),$ analytic in a neighborhood of the origin, satisfying Eq. (2) such that

$$u(x,y) = f(x,y) + \sum_{(i,j)\in \mathbf{N}^2\setminus \mathbf{I}_{mn}} e_{ij}x^i y^j. \tag{5}$$

Proof. Note that

$$\frac{\partial^{i+j}}{\partial x^i \partial y^j}\left(\sum_{l=0}^{2} a_l(x,y)u^l(x,y)\right)_{(0,0)} = 0$$

$$= \frac{\partial^{i+j}}{\partial x^i \partial y^j}\left(\sum_{l=0}^{2} a_l(x,y)f^l(x,y)\right)_{(0,0)}, (i,j)\in \mathbf{I}_{mn}.$$

Through much trivial calculation, we have

$$\left[\frac{\partial}{\partial u}\left(\sum_{l=0}^{2} a_l(x,y)u^l(x,y)\right)\frac{\partial^{i+j}u}{\partial x^i\partial y^j} + z_{ij}\right]_{(0,0)} = 0$$

$$= \left[\frac{\partial}{\partial f}\left(\sum_{l=0}^{2} a_l(x,y)f^l(x,y)\right)\frac{\partial^{i+j}f}{\partial x^i\partial y^j} + Z_{ij}\right]_{(0,0)}, \quad (i,j)\in \mathbf{I}_{mn}, \tag{6}$$

where

$$z_{i+1,j} = \frac{\partial z_{ij}}{\partial x} + \frac{\partial^{i+j}u}{\partial x^i\partial y^j}\frac{\partial}{\partial x}\left[\frac{\partial}{\partial u}\left(\sum_{l=0}^{2} a_l(x,y)u^l(x,y)\right)\right],$$

$$z_{i,j+1} = \frac{\partial z_{ij}}{\partial y} + \frac{\partial^{i+j}u}{\partial x^i\partial y^j}\frac{\partial}{\partial y}\left[\frac{\partial}{\partial u}\left(\sum_{l=0}^{2} a_l(x,y)u^l(x,y)\right)\right],$$

$$Z_{i+1,j} = \frac{\partial Z_{ij}}{\partial x} + \frac{\partial^{i+j}f}{\partial x^i\partial y^j}\frac{\partial}{\partial x}\left[\frac{\partial}{\partial f}\left(\sum_{l=0}^{2} a_l(x,y)f^l(x,y)\right)\right],$$

$$Z_{i,j+1} = \frac{\partial z_{ij}}{\partial y} + \frac{\partial^{i+j}f}{\partial x^i\partial y^j}\frac{\partial}{\partial y}\left[\frac{\partial}{\partial f}\left(\sum_{l=0}^{2} a_l(x,y)f^l(x,y)\right)\right].$$

Now, taking the unique $u(x,y)$ from Theorem 1 it is seen that since

$$\left[\frac{\partial}{\partial f}\left(\sum_{l=0}^{2} a_l(x,y)f^l(x,y)\right)\right]_{(0,0)} = \left[\frac{\partial}{\partial u}\left(\sum_{l=0}^{2} a_l(x,y)u^l(x,y)\right)\right]_{(0,0)} = \pm\sqrt{D(0,0)} \neq 0,$$

Eq. (6) with $i=1, j=0$ gives $\left.\frac{\partial f}{\partial x}\right|_{(0,0)} = \left.\frac{\partial u}{\partial x}\right|_{(0,0)},$ which with $i=2$ gives $\left.\frac{\partial^2 f}{\partial x^2}\right|_{(0,0)} = \left.\frac{\partial^2 u}{\partial x^2}\right|_{(0,0)}.$ It follows that

$$\left.\frac{\partial^i f}{\partial x^i}\right|_{(0,0)} = \left.\frac{\partial^i u}{\partial x^i}\right|_{(0,0)}, (i,0)\in \mathbf{I}_{mn}.$$

Similarly, we have

$$\left.\frac{\partial^j f}{\partial y^j}\right|_{(0,0)} = \left.\frac{\partial^j u}{\partial y^j}\right|_{(0,0)}, (0,j)\in \mathbf{I}_{mn};$$

and

$$\left.\frac{\partial^{i+j} f}{\partial x^i\partial y^j}\right|_{(0,0)} = \left.\frac{\partial^{i+j} u}{\partial x^i\partial y^j}\right|_{(0,0)}, (i,j)\in \mathbf{I}_{mn};$$

i. e. Eq. (5) holds. This completes the proof of Theorem 2.

3. NUMERICAL EXAMPLE

This paper is an attempt to answer many of the practical questions which arise when one actually tries to compute the quadratic approximation to some function. The following are two simple examples.

Example 1. Let $f(x,y)=\ln(1+x+y)$. Then

$$\left(1+\tfrac{14x}{15}-\tfrac{x^2}{5}\right)f^2(x,y)+\left(1+4x+\tfrac{16x^2}{25}\right)f(x,y)-x-\tfrac{9x^2}{2}$$
$$=\sum_{i,j=0}^{\infty}e_{ij}x^iy^j,$$

where $e_{ij}=0$ for any $(i,j)\in\mathbf{I}_{2,0}$. Also

$$u(x,y)=x-\tfrac{x^2}{2}+\tfrac{x^3}{3}-\tfrac{x^4}{4}+\tfrac{x^5}{5}-\tfrac{x^6}{6}+\tfrac{193x^7}{1350}+\cdots,$$

i.e. $$u(x,y)=f(x,y)+\sum_{i,j=0}^{\infty}e_{ij}x^iy^j,$$

where $e_{i0}=0, i=0,1,...,6$ (cf. the case $a_2(0,0)\neq 0$ in the proof of Theorem 1).

Example 2. Let

$$f(x,y)=\frac{xe^x-ye^y}{x-y}=\sum_{i,j=0}^{\infty}\tfrac{1}{(i+j)!}x^iy^j.$$

Then

$$\left(-\tfrac{11x}{240}+\tfrac{y}{12}+\tfrac{x^2}{80}-\tfrac{xy}{120}-\tfrac{x^2y}{320}\right)f^2(x,y)$$
$$+\left(1-\tfrac{8x}{15}-\tfrac{2y}{3}+\tfrac{7xy}{24}+\tfrac{x^2}{6}-\tfrac{43x^2y}{480}\right)f(x,y)$$
$$-1-\tfrac{101x}{240}-\tfrac{5y}{12}+\tfrac{41xy}{120}-\tfrac{13x^2}{240}+\tfrac{29x^2y}{960}=\sum_{i,j=0}^{\infty}e_{ij}x^iy^j,$$

where $e_{ij}=0$ for any $(i,j)\in\mathbf{I}_{2,1}$. We have

$$u(x,y)=1+x+\tfrac{x^2}{2}+\tfrac{x^3}{6}+\tfrac{x^4}{24}+\tfrac{x^5}{120}+\tfrac{x^6}{720}+y+\tfrac{xy}{2}$$
$$+\tfrac{x^2y}{6}+\tfrac{x^3y}{24}+\tfrac{x^4y}{120}+\tfrac{x^5y}{720}+\tfrac{y^2}{2}+\tfrac{xy^2}{6}+\tfrac{y^3}{6}+\tfrac{x^7}{10800}+\ldots,$$

i.e. $$u(x,y)=f(x,y)+\sum_{i,j=0}^{\infty}e_{ij}x^iy^j,$$

where $e_{ij}=0$ for any $(i,j)\in\mathbf{I}_{2,1}$. (cf. the case $a_2(0,0)=0$ in the proof of Theorem 1).

4. SEQUENCE OF BIVARIATE QUADRATIC PADE APPROXIMATION

In this section, it is shown that some "increasing" sequence of bivariate quadratic Hermite-Padé forms about one variable yields a sequence of bivariate quadratic approximations with increasing order of accuracy.

Let

$$R_x=\max\left\{r\in\mathbf{N}: f(x,y)=\sum_{i,j=0}^{\infty}e_{ij}x^iy^j,\right.$$

where $e_{i0}=0$ for any $0\le i\le r\Big\}$. then we have

Proposition 3. For any $(m,n)\in\mathbf{N}^2, m\ge R_x+1$, at least two of the vicariate coefficient polynomials $a_0(x,y), a_1(x,y), a_2(x,y)$ with degree at most m about x and n about y don't vanish identically.

Theorem 4. For any $(m,n)\in\mathbf{N}^2$, we have a quadratic approximation $u(x,y)$ to $f(x,y)$ such that

(i) $u(x,y)=f(x,y)+\sum_{i,j=0}^{\infty}e_{ij}x^iy^j,$

where $e_{i0}=0$ for any $0\le i\le q_{mn}$; and

(ii) $\lim_{m\to\infty}q_{mn}=\infty.$

Theorem 5. For any $(m,n)\in\mathbf{N}^2$, we have a quadratic approximation $v(x,y)$ to $f(x,y)$ such that

(i) $v(x,y)=f(x,y)+\sum_{i,j=0}^{\infty}e_{ij}x^iy^j,$

where $e_{0j}=0$ for any $0\le j\le q'_{mn}$; and

(ii) $\lim_{m\to\infty}q'_{mn}=\infty.$

5. REFERENCES

[1] G.A.Baker, Essentials of Padé approximants, New York: Academic Press, 1975.

[2] P. Barone, R. March, "Some properties of the asymptotic location of poles of Padé approximants to noisy rational functions, relevant for modal analysis, " IEEE Transactions on Signal Processing, 46(1998):2448-2457.

[3] D. Belki´c, "Strikingly stable convergence of the Fast Padé Transform (FPT) for high-resolution parametric and non-parametric signal processing of Lorentzian and non-Lorentzian spectra," Nuclear Instruments and Methods in Physics Research, 525(2004): 366-371.

[4] Y. Bistrits, U. Shaked, "Discrete multivariable system approximations by minimal Padé-type stable models," IEEE Transactions on Circuit System, 31(1984), 382-390.

[5] J. S. R. Chisholm, "Multivariate approximants with branch points I: off-diagonal approximants," Proceedings of The Royal Society of London, A, 362(1978): 43-56.

[6] E. Celik, "On the numerical solution of chemical differential-algebraic equations by Pade series, " Applied Numerical Mathematics, 153(2004): 13-17.

[7] R. H. Jones, G. J. Makinson, "The Generation of Chisholm Rational Approximants to Power Series in Two Variables," Institute of Mathematics with Applications Journal, 13(1974): 299-310.

Parallel GAOR Algorithms for Linear Complementarity Problem

Dongjin Yuan ,hui zhang
Department of Mathematics YangZhou University, Yangzhou, Jiangsu, China
EMAIL: dongjinyuan@yahoo.com

ABSTRACT

In order to solve a linear complementarity problem $LCP(M,q)$, When M is an n×n nonsingular matrix, a class of generalized AOR (GAOR) methods based on GAOR methods for solving linear system, whose special case reduces generalized SOR(GSOR) methods, is proposed. Some sufficient conditions for convergence of GAOR and GSOR methods are given, when the system matrix M is an *H*-matrix, *M*-matrix and a strictly or irreducible diagonally dominant matrix. When M is an *L*-matrix, their monotone convergence are discussed.

Keywords: GAOR methods; GSOR methods; Linear complementarity problem; Convergence; Monotone.

1.INTRODUCTION

The linear complementarity problem $LCP(M,q)$ consists of finding a vector *z* in the *n* dimensional space R^n such that

$$Mz+q \geq 0,\ z \geq 0,\ z^T(Mz+q)=0 \qquad (1.1)$$

Where $M \in R^{n\times n}$ and $q \in R^n$ are a given matrix and a vector , respectively. Because of the application (see[4], chapter [10]), the research on the numerical methods for solving (1.1) have got more attention.

For convergence we shall now briefly explain some of the terminology used in the next section. Let $C=(c_{kj}) \in R^{n\times n}$ be an n×n matrix. By $\mathrm{diag}(C)$ we denote the n×n diagonal matrix coinciding in its diagonal with C. For $A=(a_{kj})$, $B=(b_{kj}) \in R^{n\times n}$, we write $A \geq B$ if $a_{kj} \geq b_{kj}$ holds for all $k, j=1,2,\cdots n.$ Calling A nonnegative if $A \geq 0$, we say that $B \leq C$ if and only if $-B \geq -C$, These definitions carry immediately over to vectors by identifying them with n×1 matrix. By $|A|=(|a_{kj}|)$ we define the absolute value of $A \in R^{n\times n}$. We denote by $\langle A\rangle=(\langle a_{kj}\rangle)$ the comparison matrix of $A \in R^{n\times n}$ where $\langle a_{kj}\rangle=|a_{kk}|$ for $k=j$ and $\langle a_{kj}\rangle=-|a_{kj}|$ for $k \neq j,\ k, j=1,2,\cdots n;$. $\rho(\cdot)$ denotes the spectral radius of a matrix.

Definition 1.1.
Let $A \in R^{n\times n}$. It is called an

(1)*L*-matrix if $a_{kk}>0$ for $k=1,2,\cdots n,$ and $a_{kj} \leq 0$ for $k \neq j, k, j=1,2,\cdots n$;
(2)*M*-matrix if it is a nonsingular *L*-matrix satisfying $A^{-1} \geq 0$;
(3)*H*-matrix if $\langle A\rangle$ is an *M*-matrix.

Note that an *H*-matrix is nonsingular with the properties that $|A|^{-1} \leq \langle A\rangle^{-1}$ and $\rho(|D|^{-1}|B|) < 1$, Where $D=diag(A), B=D-A.$

Iterative methods have been found very useful for solving the complementarity problems (see[7] and reference therein). The most of these iterative methods are based on extensions of their counterparts for solving systems of linear algebraic equations (see[1,2,3,6,7,9]).

In [5], a class of generalized AOR(GAOR) methods for linear system was proposed and some convergence conditions for diagonal dominant matrix were given. While in [8], also for linear system, some sufficient and/or necessary conditions of convergence were achieved, when system matrix A is an Hermitian positive definite matrix, an *H*-, *L*- or *M*-matrix, a strictly or irreducible diagonally dominant matrix; and in [10] some sufficient conditions for parallel chaotic GAOR method for *H*-matrix were established.

In this paper, similar to [9], where the modified AOR(MAOR) methods were discussed for linear complementarity problem, we shall establish generalized AOR(GAOR) methods for solving $LCP(M,q)$ based on the models in [9]. Some sufficient conditions for convergence of the GAOR and GSOR methods will be given, when the system matrix M is an *H*-matrix , *M*-matrix and a strictly or irreducible diagonally dominant matrix. When M is an *L*-matrix, their monotone convergence are discussed.

2. NOTATION AND METHODS

If $x \in R^n$, x_+ is used to denote the vector with elements $(x_+)_j=\max\{0,x_j\}, j=1,2,\cdots n.$ For any $x, y \in R^n$, the following facts hold:

(a) $(x+y)_+ \leq x_+ + y_+$;
(b) $x_+ - y_+ \leq (x-y)_+$;
(c) $|x| = x_+ + (-x)_+$;
(d) $x \leq y$ implies $x_+ \leq y_+$.

In [1,6] it shown that the $LCP(M,q)$ can be equivalently transformed to a fixed point system of equations

$$z=(z-\alpha E(Mz+q))_+ \qquad (2.1)$$

where α is some positive constant and E is a diagonal matrix with positive diagonal elements.

The following lemma is useful in this paper.

Lemma 2.1 (see[9]). Let $M \in R^{n\times n}$ be an *H*-matrix with positive diagonal elements. Then the $LCP(M,q)$ in (1.1)has a unique solution $z^* \in R^n$.

In the case when $D = diag(M)$ is nonsingular, as some paper (see[9]) mentioned, a particular but more practical choice for α and E is $\alpha E = D^{-1}$. Then (2.1) is reduced to

$$z = (z - D^{-1}(Mz + q))_+ \qquad (2.2)$$

In this case, Let the matrix M be split as

$$M = D + L + U, \qquad (2.3)$$

Where $D = diag(M)$, L and U are strictly lower and upper triangular matrices, respectively. Then from (2.2), we define the AOR method for $LCP(M,q)$ as follows:

$$z^{p+1} = (z^p - D^{-1}[\gamma L z^{p+1} + (\omega M - \gamma L)z^p + \omega q])_+ \qquad (2.4)$$

When $\gamma = \omega$ the AOR method reduces to SOR method. In the splitting (2.3), we always assume that D is nonsingular and denote

$$J = D^{-1}(L + U) \qquad (2.5)$$

Then, for $\Omega = \text{diag}(\omega_1, \cdots \omega_n)$ with $\omega_i \in R_+$ and α, a real parameter, a generalized AOR (GAOR) method for solving the $LCP(M,q)$ can be described as follows:

GAOR method:_

Step 1. Choose an initial vector $z^0 \in R^n$ and set $p := 0$.

Step 2. For $p = 0,1,2,\cdots$, let z^{p+1} be calculated according to

$$z^{p+1} = (z^p - D^{-1}[\alpha\Omega L z^{p+1} + (\Omega M - \alpha\Omega L)z^p + \Omega q])_+ \qquad (2.6)$$

Step 3. If $z^{p+1} = z^p$, then stop. Otherwise, set $p := p + 1$ and return to step 2.

Remark 2.1. In order to get more convenient formula , we denote $M = (m_{kj}), z^p = (z_1^p \cdots z_n^p)^T$ and $q = (q_1, \cdots q_n)^T$, then (2.6) can be rewritten as

$$z_k^{p+1} = ((1-\omega_k)z_k^p - \frac{\omega_k}{m_{kk}}[\alpha\sum_{j=1}^{k-1} m_{kj}(z_j^{p+1} - z_j^p) + \sum_{j=1, j\neq k}^{n} m_{kj} z_j^p + q_k])_+ \quad k = 1,2,\cdots n. \qquad (2.7)$$

i.e., $z_k^{p+1} = 0$, if

$$(\omega_k - 1)z_k^p + \frac{\omega_k}{m_{kk}}[\alpha\sum_{j=1}^{k-1} m_{kj}(z_j^{p+1} - z_j^p) + \sum_{j=1, j\neq k}^{n} m_{kj} z_j^p + q_k] > 0$$

and otherwise

$$z_k^{p+1} = (1-\omega_k)z_k^p - \frac{\omega_k}{m_{kk}}\alpha\sum_{j=1}^{k-1} m_{kj}(z_j^{p+1} - z_j^p) - \frac{\omega_k}{m_{kk}}\sum_{j=1, j\neq k}^{n} m_{kj} z_j^p - \frac{\omega_k}{m_{kk}} q_k .$$

When α =1 the GAOR method reduces to GSOR method. Therefore, we can define the GSOR method for $LCP(M,q)$ in the following:

GSOR method:

Step1: Choose an initial vector $z^0 \in R^n$ and set $p := 0$.

Step2: For $p = 0,1,2,\cdots$, let z^{p+1} be calculated according to

$$z^{p+1} = (z^p - D^{-1}[\Omega L z^{p+1} + (\Omega D + \Omega U)z^p + \Omega q])_+ \qquad (2.8)$$

Step3: If $z^{p+1} = z^p$, then stop. Otherwise, set $p := p + 1$ and return to step 2.

Remark 2.2. Similar to the GAOR method, (2.8) can be rewritten as

$$z_k^{p+1} = ((1-\omega_k)z_k^p - \frac{\omega_k}{m_{kk}}[\sum_{j=1}^{k-1} m_{kj} z_j^{p+1} + \sum_{j=k+1}^{n} m_{kj} z_j^p + q_k])_+, \; k = 1,2,\cdots n \qquad (2.9)$$

or equivalently, $z_k^{p+1} = 0$, if

$$(\omega_k - 1)z_k^p + \frac{\omega_k}{m_{kk}}[\sum_{j=1}^{k-1} m_{kj} z_j^{p+1} + \sum_{j=k+1}^{n} m_{kj} z_j^p + q_k] > 0$$

and otherwise

$$z_k^{p+1} = (1-\omega_k)z_k^p - \frac{\omega_k}{m_{kk}}[\sum_{j=1}^{k-1} m_{kj} z_j^{p+1} + \sum_{j=k+1}^{n} m_{kj} z_j^p + q_k].$$

3.CONVERGENCE ANALYSIS FOR *H*--MATRIX

At first we define the operator $f : R^n \to R^n$, in accordance with the rule: $f(z) = \xi$,where ξ is the fixed point of the system of equations

$$\xi = (z - D^{-1}[\alpha\Omega L\xi + (\Omega M - \alpha\Omega L)z + \Omega q])_+$$

Then by denoting

$$Q = I - \alpha D^{-1}\Omega L, R = I - D^{-1}(\Omega M - \alpha\Omega L),$$

We can prove the following convergence theorem for the GAOR method.

Theorem 3.1

Let $M = (m_{kj}) \in R^{n\times n}$ be an *H*-matrix with positive diagonal elements and let the splitting (2.3) satisfy $\langle M \rangle = D - |L| - |U|$ .If $0 < \omega_i < 2/(1+\rho(|J|)), i = 1,2,\cdots n,$ and $\alpha \in R_+$ then for any initial guess $z^0 \in R^n$, the iterative sequence $\{z^p\}$ generated by the GAOR method converges to the unique solution z^* of the $LCP(M,q)$ and

$$\rho(\langle Q\rangle^{-1}|R|) \le \max_{1\le i\le n}\{|1-\omega_i| + \omega_i\rho(|J|)\} < 1, \tag{3.1}$$

whenever $0 < \alpha \le 1$, and

$$\rho\left(\langle Q\rangle^{-1}|R|\right) \le \left|1-\frac{1}{\alpha}\right| + \frac{1}{\alpha}\max_{1\le i\le n}\{|1-\omega_i| + \omega_i\rho(|J|)\} < 1. \tag{3.2}$$

whenever $\alpha \ge 1$.

As the special case, for the GSOR method, we have the following convergence results.

Corollary 3.2. Let $M = (m_{kj}) \in R^{n\times n}$ be an *H*-matrix with positive diagonal elements and let the splitting (2.3) satisfy

$$\langle M \rangle = D - |L| - |U|.$$

If $0 < \omega_i < 2/(1+\rho(|J|)), i = 1,2,\cdots n,$

Then, for any initial guess $z^0 \in R^n$,the iterative sequence $\{z^p\}$ generated by the GSOR method converges to the unique solution z^* of the $LCP(M,q)$ and

$$\rho(\langle Q\rangle^{-1}|R|) \le \max_{1\le i\le n}\{|1-\omega_i| + \omega_i\rho(|J|)\} < 1,$$

Corollary 3.3. Let $M \in R^{n\times n}$ be strictly diagonally dominant by rows with positive diagonally elements. Then for any initial guess $z^0 \in R^n$, the iterative sequence $\{z^p\}$, $p = 0,1,2,\cdots$, generated by the GAOR method or the GSOR method converges to the unique solution z^* of $LCP(M,q)$ and

$$(a)\rho(\langle Q\rangle^{-1}|R|) \le \max_{1\le i\le n}\{|1-\omega_i| + \omega_i\sigma\} < 1 \tag{3.3}$$

whenever $0 < \omega_i < \frac{2}{1+\sigma}, i = 1,2,\cdots n, 0 < \alpha \le 1,$

for GAOR method;

$$(b)\rho(\langle Q\rangle^{-1}|R|) \le \left|1-\frac{1}{\alpha}\right| + \frac{1}{\alpha}\max_{1\le i\le n}\{|1-\omega_i| + \omega_i\sigma\} < 1$$

whenever

$$0 < \omega_i < \frac{2}{1+\sigma}, i = 1,2,\cdots n, \alpha \ge 1,$$

for GAOR method;

$$(c)\rho(\langle Q\rangle^{-1}|R|) \le \max_{1\le i\le n}\{|1-\omega_i| + \omega_i\sigma\} < 1, \tag{3.4}$$

whenever

$$0 < \omega_i < \frac{2}{1+\sigma}, i = 1,2,\cdots n,$$

for GSOR method.

Remark 3.4.

When M is irreducible diagonally dominant by rows, the inequalities $\omega_i < 2/(1+\sigma)$, $i = 1,2,\cdots n$, in the above corollary can be replaced by $\omega_i \le 2/(1+\sigma), i = 1,2,\cdots n$. Then for this case, the other convergence results in corollary 3.3 can be changed similarly. (see [8]).

4. MONOTONE CONVERGENCE ANALYSIS

In this section, we investigate the monotone convergence properties of the GAOR and GSOR method when the system matrix $M \in R^{n\times n}$ is an *L*-matrix. It will show that, for some initial guesses, the iterative sequence generated by the GAOR method or the GSOR method converges towards a solution of the $LCP(M,q)$ for above. For this purpose, we define a set

$$\Delta = \{x \in R^n \mid x \ge 0, Mx + q \ge 0\}$$

Clearly, if the $LCP(M,q)$ is solvable, then the set Δ is nonempty.

At first, we study the monotone properties of the operator $f : R^n \to R^n$ with $f(z) = \xi$,

$$\xi = (z - D^{-1}[\alpha\Omega L\xi + (\Omega M - \alpha\Omega L)z + \Omega q])_+ \tag{4.1}$$

Theorem 4.1

Let the operator $f : R^n \to R^n$ be defined in (4.1). Assume that $M \in R^{n\times n}$ is an *L*-matrix, and it has the splitting (2.3). Also, assume that $0 < \omega_i \le 1$, $i = 1,2,\cdots n, 0 \le \alpha \le 1$ .Then for any $z \in \Delta$, it holds that:

$(a)\ f(z) \le z;$

$(b)\ y \le z$ implies $f(y) \le f(z)$;

$(c)\ \xi = f(z) \in \Delta$;

Proof.

It is easy to show that for $z \in R_+$, $z \in \Delta$ if and only if

$$Dz + Lz + Uz + q \ge 0,$$

i.e.,

$$m_{kk}z_k + \sum_{j=1}^{k-1} m_{kj}z_j + \sum_{j=k+1}^{n} m_{kj}z_j + q_k \geq 0$$
$$k = 1,2,\cdots n. \quad (4.2)$$

We first verify (a). Let $\xi = f(z) = (\xi_1, \xi_2, \cdots, \xi_n)^T$ by GAOR method it gets

$$\xi_1 = (z_1 - \frac{\omega_1}{m_{11}}[\sum_{j=1}^{n} m_{kj}z_j + q_k])_+,$$
$$\xi_2 = (z_2 - \frac{\omega_2}{m_{22}}[\alpha m_{21}(\xi_1 - z_1) + \sum_{j=1}^{n} m_{kj}z_j + q_2])_+,$$
$$\xi_k = (z_k - \frac{\omega_k}{m_{kk}}[\alpha \sum_{j=1}^{k-1} m_{kj}(\xi_j - z_j) + \sum_{j=1}^{n} m_{kj}z_j + q_k])_+,$$
$$\xi_n = (z_n - \frac{\omega_n}{m_{nn}}[\alpha \sum_{j=1}^{n-1} m_{kj}(\xi - z_j) + \sum_{j=1}^{n} m_{kj}z_j + q_n])_+. \quad (4.3)$$

Hence, using (4.2) and $D > 0$, from the first equality in (4.3) it follows that $\xi_1 \leq z_1$. And from the second equality to the last one in (4.3),also by (4.2) and $D > 0, L \leq 0,$ it gives $\xi_2 \leq z_2, \cdots, \xi_k \leq z_k, \cdots \xi_n \leq z_n$. Then, $\xi = f(z) \leq z$ and this shows (a).

To verify (b), we denote $\eta = f(y) = (\eta_1, \eta_2, \cdots \eta_n)^T$. By GAOR method, it gets

$$\eta_k = (y_k - \frac{\omega_k}{m_{kk}}[\alpha \sum_{j=1}^{k-1} m_{kj}(\eta - y_j) + \sum_{j=1}^{n} m_{kj}\eta_j + q_k])_+, k = 1,2,\cdots n. \quad (4.4)$$

Then by subtracting (4.3) from (4.4) we have

$$\eta_k - \xi_k \leq ((y_k - z_k) - \frac{\omega_k}{m_{kk}}[\alpha \sum_{j=1}^{k-1} m_{kj}(\eta_j - \xi_j - (y_j - z_j)) + \sum_{j=1}^{n} m_{kj}(y_j - z_j)])_+$$
$$= [(1-\omega_k)(y_k - z_k) - \frac{\omega_k}{m_{kk}}[\alpha \sum_{j=1}^{k-1} m_{kj}(\eta_j - \xi_j) + (1-\alpha)\sum_{j=1}^{k-1} m_{kj}(y_j - z_j) + \sum_{j=k+1}^{n} m_{kj}(y - z_j)]_+$$
$k = 1,2,\cdots n$.

Since $1-\omega \geq 0, 1-\alpha \geq 0, y \leq z$ and $L \leq 0, U \leq 0$, we see that $\eta_1 \leq \xi_1$,and from this, it gets

$$\eta_2 \leq \xi_2, \cdots \eta_n \leq \xi_n, i.e., \eta \leq \xi, f(y) \leq f(z).$$

Now, we turn to (c). By the definition of operator f we have $\xi = f(z) \geq 0$. On the other hand, by the definition $(\bullet)_+$ and from (4.3),we can obtain

$$\begin{aligned} M\xi + q &= D\xi + L\xi + U\xi + q \\ &\geq D\xi + L\xi + Uz + q \\ &\geq D(z - D^{-1}\Omega[\alpha L(\xi - z) + Mz + q] \\ &\quad + L\xi + Uz + q \\ &= Dz - \alpha\Omega L(\xi - z) - \Omega(D + L + U)z \\ &\quad - \Omega q + L\xi + Uz + q \\ &= (I - \Omega)(Dz + Uz + q) \\ &\quad + (I - \alpha\Omega)L\xi - (1-\alpha)\Omega Lz \\ &\geq (I - \Omega)(Dz + Uz + q) \\ &\quad + (I - \alpha\Omega)Lz - (1-\alpha)\Omega Lz \\ &= (I - \Omega)(Dz + Uz + q) + (I - \Omega)Lz \\ &= (I - \Omega)(Dz + Lz + Uz + q) \\ &\geq 0, \end{aligned}$$

and consequently, $\xi \in \Delta$.

Using Theorem 4.1, we will prove the following monotone convergence about the GAOR and GSOR methods.

Theorem 4.2.

Assume that $M \in R^{n\times n}$ is an *L*-matrix. Also, assume that $0 < \omega_i \leq 1, \quad i = 1,2\cdots n, \quad 0 < \alpha \leq 1$,Then for any initial vector $z^0 \in \Delta$, the iterative sequence $\{z^p\}, p = 1,2,\cdots$, generated by the GAOR method or the GSOR method has the following properties:

(a) $0 \leq z^{p+1} \leq z^p \leq z^0, p = 0,1,2,\cdots;$

(b) $\lim_{p\to\infty} z^p = z^*$ is the unique solution of the $LCP(M,q)$.

Proof.

We only give the proof for the GAOR method. Since $z^0 \in \Delta$,by (a) of Theorem 4.1 we have $z^1 \leq z^0$ and $z^1 \in \Delta$. Now, using (a) of Theorem 4.1 again, by recursively we have shown that validity of (a).

The inequalities given in (a) show that the sequence $\{z^p\}$, $p = 1,2,\cdots$, is monotone bounded, so that it converges to some vector z^* satisfying

$$z^* = (z^* - D^{-1}[\alpha\Omega L z^* + (\Omega M - \alpha\Omega L)z^* + \Omega q])_+$$

i.e.,

$$z^* = (z^* - D^{-1}[\Omega M z^* + \Omega q])_+ .$$

Hence, z^* is the unique solution of the $LCP(M,q)$.

In the following we discuss the influences of the parameters $\omega_i, i = 1,2,\cdots n$ and α upon the convergence rate of the GAOR and GSOR methods.

Theorem 4.3.

Let $M \in R^{n\times n}$ be an *L*-matrix. Then, for any initial vector $z^0 = y^0 \in \Delta$,both the iterative sequences $\{z^p\}$ and $\{y^p\}$, $p = 0,1,2,\cdots$,generated by the GAOR (or GSOR) method corresponding to the parameters $(\alpha, \omega_1, \cdots \omega_n)$ and $(\overline{\alpha}, \overline{\omega}_1, \cdots \overline{\omega}_n)$, respectively, converge to the solution $z^* \in R^n$ of $LCP(M,q)$ and it holds

$$z^p \le y^p, p = 0,1,2,\cdots, \quad (4.5)$$

provided the parameters $(\alpha, \omega_1, \cdots \omega_n)$ and $(\overline{\alpha}, \overline{\omega}_1, \cdots \overline{\omega}_n)$ satisfy

$$0 < \overline{\omega}_i \le \omega_i \le 1, i = 1,2,\cdots n, 0 < \overline{\alpha} \le \alpha \le 1.$$

Remark 4.4.

Theorem4.2 and 4.3 show that the parameter collections $\omega_1 = \cdots \omega_n = \alpha = 1$ can result in faster convergence rate of the GAOR and GSOR methods under the assumptions. This also implies that the optimum parameters, in general, should be $\alpha^*, \omega_1^*, \cdots \omega_n^* \in [1, \infty)$.

References

[1]B.H.Ahn, Solution of nonsymmetric linear complementarity problems by iterative methods, J.Optim. Theory Appl.33(1981) 175-185.

[2]Z.Z.Bai, On the monotone convergence of matrix multisplitting relaxation methods for the linear complementarity problem, IAM J. Numer. Anal.18(1998) 509-518.

[3]Z.Z.Bai, D.J. Evans, Chaotic iterative methods for linear complementarity problem, J. Comput. Appl. Math. 63(1997) 127-138.

[4]A. Berman, R.J. plemmon, Nonnegative Matrix in the Mathematical Sciences, Academic Press, New York, 1979.

[5]K.R.James, Convergence of matrix iterations subject to diagonal dominance,SIAM J. Numer. Anal. 12(1973) 478-484.

[6]O.L. Mangasarian, Solution of symmetric linear complementarity problems by iterative methods, J. Optim. Theory Appl.22(1997) 465-485.

[7]J.S. Pang, Necessary and sufficient conditions for the convergence of iterative methods for the linear complementarity problem, J. Optim. Theory Appl. 42(1984) 1-17.

[8]Y.Z.Song, On the convergence of the generalized AOR method, Linear. Algebra Appl. 256(1997) 199-218.

[9]D.J.Yuan, Y.Z.Song, Modified AOR methods for linear complementarity problem, Appl. Math. Comput. 140(2003) 53-67.

[10]D.J.Yuan, Convergence of parallel chaotic generalized AOR method for *H*-matrix, DCABES 2001 proceedings 24-28

Parallel Evolution Finite Element Method on PC Cluster*

Zhongliang Ru Hongbo Zhao
College of Civil Engineering, Henan Polytechnic University
Jiaozuo, Henan 454000, China
Email: ruzl@hpu.edu.cn
Mingtian Li
Department of Civil engineering, Shandong Jiaotong University
Jinan , Shandong 250023, China
Email: limt@hotmail.com

ABSTRACT

A parallel evolution FEM algorithm based on PC cluster was designed by combining Genetic Algorithms(GAs) and domain decomposed parallel finite element method. First generate a random population by GAs, then exert the virtues of parallel FEM huge computing capacity, fast computing speed, after the parallel FEM positive analysis, compute the fitness which was the square sum of the difference between FEM analysis and in-site monitor of the key points displacement, then a group of new parameters were generated again by the operations of GAs, the operations would done until the best parameters have been founded. This algorithm conquers the long time consuming disadvantage of GAs. Based on this algorithm developed the parallel computing program GAPFEM^{3D} using C++ and MPI library on PC cluster. GAPFEM^{3D} exerts both GA's attribute: large search space, quickly converge and parallel FEM's attribute: large scale capability, high efficiency, finally applied it to large-scale geoengineering successfully. The back analysis results validated the robust and efficiency of this algorithm.

Keywords: PC Cluster, Domain Decomposed Method, Genetic Algorithms.

1. INTRODUCTION

It is well known that there are many inherent uncertain factors in the geoengineering such as material properties, geometry or loading conditions. How to get the proper parameters of the real geo-engineering becomes a focus to all the engineers[1]. Displacement back analysis is a popular method in the rock engineering problems. The main objective of it is to determine the initial stresses and material constants etc. from the field measurement data. Back analysis problems may be solved in two different ways, defined as inverse and direct approaches. In the inverse approach, the mathematical formulation is just the reverse of ordinary stress analysis. It numerically solves some of the material parameters or loading conditions based on measured displacements. Rapid numerical solution is one of the advantages of the inverse method. The direct approach is based on an iterative procedure correcting the trial values of unknown parameters by minimizing error functions; hence, no formulation of the inverse problem is required. Standard algorithms of mathematical programming, such as Simplex or Rosenbrock method, proved to be very efficient for the small-scale problems. But for the large-scale problems the iterative procedure requires high computational effort and long time consuming. It becomes an obstacle to handle its application[2-3]. Aim to this disadvantage, this paper combined Genetic Algorithms(GAs) and parallel finite element method, then designed a genetic parallel finite element method (GAPFEM) algorithm for engineering's back analysis.

GAPFEM^{3D} program is developed on PC cluster using C++ and MPI library, the parallel FEM algorithm is based on the Domain Decomposed Method (DDM). It decomposed the FEM mesh (big problem) to several submeshs (small problems) then distributed them to the PCs, analyzed those small problems simultaneous. So it can analysis big-scale engineering and shorten the computing time by dividing the big problem to small problems and computing it by several PCs. This paper is organized as follows: in Section 2 the architecture of PC cluster is outlined, the theories of DDM method on PC cluster is presented. In Section 3 parallel implementation of the GAPFEM algorithm is described and in Section 4 a numerical example is given. Finally, the paper is summarized in Section 5.

2. PARALLEL FEM BASED ON DDM

2.1 PC Cluster

Traditionally parallel computing was very expensive. Large specialized parallel computers are costly to operate and to maintain. Generally researchers in universities and small institutes have difficulty in accessing these high performance computing resources. With the performance improvements and price reduction of PCs and the availability of new, high speed interconnection technologies, many people attempted to use them for parallel applications. In 1994 a machine composed of microprocessors connected by channel bonded Ethernet, gave place to the so-called Beowulf class cluster computers [4,5]. The Beowulf project proved that a collection of desktop PCs connected by a fast internal network can deliver a competitive performance at a very low cost[6,7]. Different from Beowulf, we built our parallel computer based on PC cluster on Windows in June, 2000. In what follows, parallel systems in hardware and software are described.

PC cluster is a set of PC-based systems interconnected through an Ethernet. Figure 1 illustrates the architecture of our PC cluster. It was built from 10 separate desktop PCs. Among these PCs, one is used as master where master process located the others as slavers. These computers are connected together with a fast Ethernet network by a 100Mb/s 3Com SuperStack 3300 switch in full duplex mode. Each computer consists of a Pentium IV 1.6GHz CPU, an IDE hard disk with 4GB capacity, a 256 Mb memory, and

* This research was supported by Henan Polytechnical Doctor Foundation(648508)

I/O devices. The cache size is 256Kb.

There are three types of software that must be installed to make the parallel system run: operating system, programming language and tools for parallel programming. In most cases the workstations ran some kind of Unix OS, and most of the message passing libraries (PVM, MPI) was developed oriented toward the Unix world. In our PC cluster, each node runs its own operating system, like Microsoft Windows NT/2000/XP. The programming languages intended for use are FORTRAN and C/C++. Microsoft Visual C++ 6.0 and digital Visual Fortran 6.0 were installed. For parallel programming tool, we use Message Passing Interface (MPI), which has become the most popular technique for implementing parallel applications on distributed memory machines. The MPI implementation used is the MPICH1.2 .

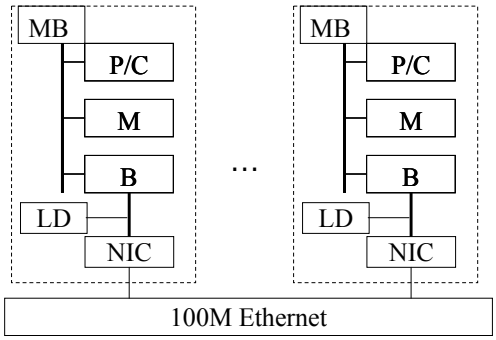


Fig.1 Architecture of PC cluster

2.2 Domain Decomposed Method

After more than four decades of intensive development, the Finite Element Method (FEM) has become an effective and widely used tool in geotechnical engineering because of its ability to model complex structural geometries, boundary and loading conditions, and material properties. Traditional FEM code is designed by series idea for single PC, from building FEM model, assembling control equations, solving the equations, data inputting/outputting are programming by process schedule. In recent years, the scale and complexity of engineering computations have increased significantly. And it makes the FE simulations tend to be very time-consuming. A FE computation of several weeks is quite common.

DDM as an idea for solving parallel FEM computation was developed to achieve better efficiency and performance[8-12]. Partitioning a FE model into subdomains is often employed to reduce the analysis effort required for one large problem into several manageable interconnected smaller problems. The method partitioning the physical domain is particularly useful in the context of distributed computations where each subdomain can be assigned to one of many processors acting in parallel. Each of PCs can achieve data inputting/outputting, element stiffness matrix computing, assembling global stiffness matrix, load vector assembling, and displacement condition handling etc simultaneous, each task is more independent and parallel; DDM on PCs can get higher parallel efficiency than series FEM[Fig.2].

2.3 PCG Solver

The finite element method is used for the approximate solution of partial differential equations. The differential equation is discretized into a series of finite element equations that form a system of equations to be solved.

$$Ku = f \tag{1}$$

Where K is stiffness matrix; u is nodal displacement vector; f is applied load vector.

But in DDM algorithm after the operations in section 2.2, the global control equation[Eq.(1)] of FEM domain can be written as distributed form[Eq.(2)] on each PC[13]. The subdomain's displacement degrees of freedom $\{u\}$ are divided into interior degrees of freedom $\{u_i\}$ and boundary displacement degrees of freedom $\{u_b\}$. In a similar way the subdomain's load degrees of freedom $\{f\}$ is divided into interior load degrees of freedom $\{f_i\}$ and load boundary degrees of freedom $\{f_b\}$[14].

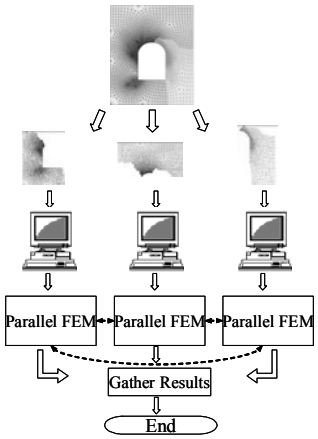


Fig.2 Sketch of FEM based on DDM

$$\begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{bmatrix}_j \begin{Bmatrix} u_i \\ u_b \end{Bmatrix}_j = \begin{Bmatrix} f_i \\ f_b \end{Bmatrix}_j \tag{2}$$

For the parallel solution of the linear system a common practice is to use some iterative method like Conjugate Gradient (CG) method. The CG algorithm is a Krylov subspace method used to solve the symmetric positive definite linear system. As is known, the convergence rate of CG depends on the spectral condition number of the coefficient matrix K. Furthermore the condition number of the matrix A generally increases with the dimension of the system. So for ill-conditioned linear systems, it is often necessary to use a preconditioning technique that accelerates convergence. This is achieved by solving the following equivalent problem with better spectral properties[15,16].

The PCG algorithm[17] for solving Eq(2) is presented as follow:

1) b_b=Update(b_b)
2) $x_i^{(0)} = 0,\ x_b^{(0)} = 0$
3) $r_i^{(0)} = b_i,\ r_b^{(0)} = b_b$
4) Solve $M(z_{\mathrm{i}}^{(0)}, z_{\mathrm{b}}^{(0)}) = (r_{\mathrm{i}}^{(0)}, r_{\mathrm{b}}^{(0)})$
5) $p_i^{(0)} = z_i^{(0)},\ p_b^{(0)} = z_b^{(0)}$
6) $\gamma^{(0)} =$ inner product $(z_i^{(0)}, z_b^{(0)}, r_i^{(0)}, r_b^{(0)})$

For k=0，1，2，… cycle

7) $(q_i^{(k)}, q_b^{(k)}) =$ matrix multiple vector $(A, p_i^{(k)}, p_b^{(k)})$
8) $\tau^{(k)} =$ inner product $(p_i^{(k)}, p_b^{(k)}, q_i^{(k)}, q_b^{(k)})$
9) $\alpha^{(k)} = \gamma^{(k)} / \tau^{(k)}$
10) $x_i^{(k+1)} = x_i^{(k)} + \alpha_k p_i^{(k)},\ x_b^{(k+1)} = x_b^{(k)} + \alpha_k p_b^{(k)}$

11) $r_i^{(k+1)} = r_i^{(k)} + \alpha_k q_i^{(k)}$, $r_b^{(k+1)} = r_b^{(k)} + \alpha_k q_b^{(k)}$

12) Solve $M(z_i^{(k+1)}, z_b^{(k+1)}) = (r_i^{(k+1)}, r_b^{(k+1)})$

13) $\gamma^{(k+1)} =$ inner product $(z_i^{(k+1)}, z_b^{(k+1)}, r_i^{(k+1)}, r_b^{(k+1)})$

14) If（$\sqrt{\gamma^{(k+1)}} <$ tolerance）exit

15) $\beta^{(k)} = \gamma^{(k+1)} / \gamma^{(k+1)}$

16) $p_i^{(k+1)} = r_i^{(k+1)} + \beta^{(k)} p_i^{(k)}$,
$p_b^{(k+1)} = r_b^{(k+1)} + \beta^{(k)} p_b^{(k)}$

With regard to matrix-vector multiplication there may be a problem in distributed memory machines since each processor has only a segment of the vector in its memory. So we need communication for other elements of the vector. In domain decomposition method, these elements locate in the neighbouring subdomain. Each processor only requires the values in the neighbouring subdomain, and at the same time send the relative values to neighbouring subdomains. In the case of DDM, the solution of the auxiliary preconditioner system can be solved independently. Then the subdomain solutions are added in order to obtain the global solution[18].

3. EVOLUTION PARALLEL FEM ALGORITHM

3.1 Genetic Algorithm

GA is probably the best-known evolution algorithm, receiving substantial attention in recent years. The first attempt to use it took place in the sixties by a team of biologists and was focused in building a computer program that would simulate the process of evolution in nature. However, the GA model used in this study and in many other structural design applications refers to a model introduced and studied by Holland[19] and co-workers. In general the term genetic algorithm refers to any population-based model that uses various operators (selection – crossover – mutation) to evolve. In the basic genetic algorithm each member of this population will be a binary or a real valued string, which is sometimes referred to as a genotype or, alternatively, as a chromosome[20].

There are three main steps of the basic GAs:

1) Initialization, The first step in the implementation of any genetic algorithm is to generate an initial population. In most cases the initial population is generated randomly. In this study in order to perform a comparison between various optimization techniques the initial population is fixed and is chosen in the neighborhood of the initial design used for the mathematical programming method. After creating an initial population, each member of the population is evaluated by computing the representative objective and constraint functions and comparing it with the other members of the population.

2) selection: Selection operator is applied to the current population to create an intermediate one. In the first generation the initial population is considered as the intermediate one, while in the next generations this population is created by the application of the selection operator.

3) generation (crossover – mutation): In order to create the next generation, crossover and mutation operators are applied to the intermediate population to create the next population. Crossover is a reproduction operator, which forms a new chromosome by combining parts of each of the two parental chromosomes. Mutation is a reproduction operator that forms a new chromosome by making (usually small) alterations to the values of genes in a copy of a single parent chromosome. The process of going from the current population to the next population constitutes one generation in the evolution process of a genetic algorithm. If the termination criteria are satisfied the procedure stops, otherwise, it returns to step 1.

3.2 Evolution Parallel FEM Algorithm

For the sake of shorten back analysis computing time, combined with 3D parallel elastic-plastic FEM code developed previously; this paper designed a master-slave parallel evolution FEM algorithm base on PC cluster. First the master PC_0 generate a random population by GA, then broadcast it's individual to all the slaver PCs, when the slavers receive the individual, begin the parallel FEM analysis immediately with the master PC_0, after all the PCs computed the fitness, sent it to the master PC_0, the master stored it. When all the individuals have been computed, the GA make selection and mutation operations, generated the next population, and then begin the next evolution, till to the results satisfied the criteria. The computing flow chat is shown as Fig.3.

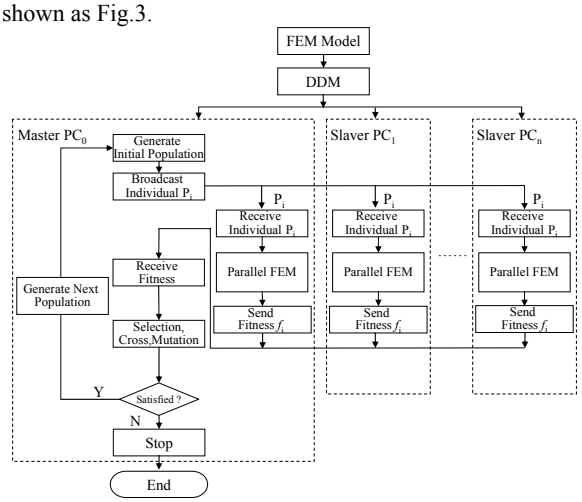


Fig.3 Flow chat of the evolution parallel finite element method algorithm

3.3 Fitness Function

A fitness function is used to determine the suitability of the material parameters for a engineering in the back analysis with a genetic algorithm. Since a fitness value is used to search for a proper parameter, that is, the fitness individual, information about the benchmark points should be included in the function. For each genetic individual the FEM calculate a benchmark point displacement value μ_i, then determine the error between the real value u_i using Eq(3).

$$\delta u_i = u_i - \mu_i \tag{3}$$

For all the benchmark points the fitness function can be express as follow:

$$f_{fitness} = \sqrt{\frac{\delta u_1^{\,2} + \delta u_2^{\,2} + \cdots + \delta u_i^{\,2} + \cdots + \delta u_N^{\,2}}{N}} \tag{4}$$

in which N is the number of benchmark points.

But in the evolution procedure based on domain decomposed method, the benchmark points may be distributed on many subdomains, so the fitness function[Eq(4)] should be modified. For example, a domain

was decomposed to M subdomains, subdomain j has N_j benchmark points, the fitness of this subdomain is obtained using following equation:

$$f_{fitness}^{j} = \delta u_1^{\ 2} + \delta u_2^{\ 2} + \cdots + \delta u_i^{\ 2} + \cdots + \delta u_{Nj}^{\ 2} \quad (5)$$

In which, if benchmark point i is shared with s subdomains, its error amount may be calculated as follow:

$$\delta u_i^2 = \delta u_i^2 / S \quad (6)$$

The whole domain's fitness is determined by the follow equation:

$$f = \sqrt{\frac{\sum_{j=1}^{M} f_{fitness}^{j}}{N}} \quad (7)$$

3.4 Data Communication

Form the algorithm above, we can see in this parallel analysis flow, at one sample simulation's start the master PC0 must send the individual's data to all the slaver, at the end of parallel FEM, the slavers sent the fitness data to master as well as. In this GAPFEM code, the communication function is implemented by MPI function's PACK and UNPACK operation, the master "pack" the data to a continuous buffer before sending, the slavers "unpack" the data after received the package.

Pack function: MPI_PACK(inbuf, incount, datatype, outbuf, outcount, position, comm)

Unpack function: MPI_UNPACK(inbuf, insize, position, outbuf, outcount, datatype, comm)

In the same way, after the slavers computing the fitness data of its subdomain, pack the data and send the package to the master. The master receives the package and unpack it, analysis the fitness then begin the next evolution. Using the PACK/UNPACK technique can enhance the entireness of the communicated data, shorten the communication consuming, and enhance the code's readability and expansibility.

4. CASE STUDY

A tunnel[Fig.4] with inner radius 1m, outer radius 10m, initial stress $\sigma_x = \sigma_y$=-30MPa, rock material parameter are E=6.7779GPa, υ =0.21034，C=3.45MPa， ϕ=30° . Select the 19 points along the inner radial lines as displacement benchmark. The range of elastic model is 1-15GPa; GA parameters are population 30, generation 30, cross ratio 0.98, mutation ratio 0.02.

Using the GAPFEM program to analysis the problem, the whole mesh is decomposed to 2-4 subdomains [Fig.5], mapping them to 2-4 PCs respectively, and compare the parallel efficiency of different PCs. The elastic and elasto-plastic analysis time verse different number of PC is shown in Fig.6.

The parallel speedup of performance is used to evaluate the time saving achieved by the parallel computing framework and the impact of the number of PC on the framework performance[21]. This measure compares the performance of the GAPFEM on a number of PCs to that a serial GAFEM on a single PC and can be calculated using Eq(8).

$$S_p = \frac{T_s}{T_p} \quad (8)$$

Where S_p—parallel speedup; T_s—elapsed time of serial GAFEM on single PC; T_p—elapsed time of GAPFEM on a cluster of p PCs.

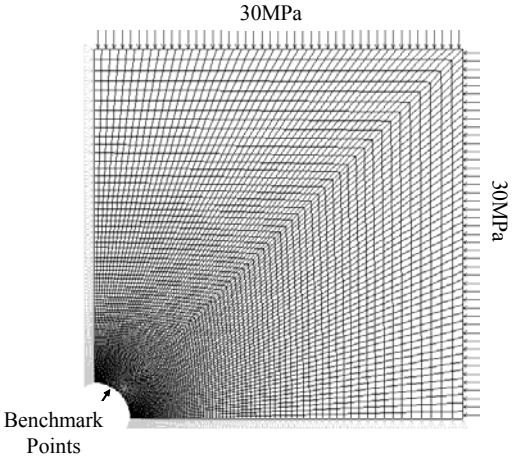


Fig.4 FEM model

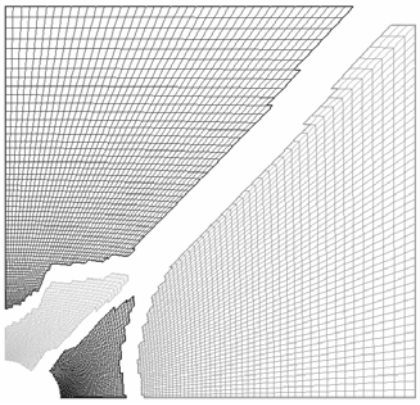


Fig. 5 FEM model after domain decomposed

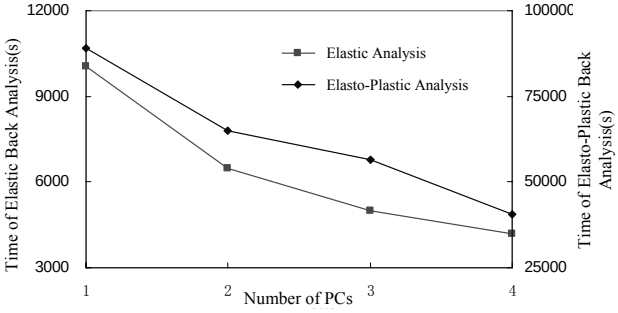


Fig. 6 Elapsed time verse PCs

In this case, using 1-4 PC to back analysis the rock's elastic model, clocking the time of each simulation, the speedup are list in table 1. This results show that the utilization of a reasonable number of PCs in a parallel computing framework can produce significant speedup and time saving for a engineering problem.

Table 1 Speed-up of parallel computing

Analysis Type	Subdomains			
	1	2	3	4
Elastic Back Analysis	1	1.549	2.008	2.386
Elasto-Plastic Back Analysis	1	1.371	1.576	2.192

5. CONCLUSION

In this paper a new evolution parallel FEM algorithm has been presented, implemented using C++ and MPI library on PC cluster, then applied this method to a tunnel. The following conclusions can be drawn: (1) Combined GAs and parallel FEM can exert the big-scale problems, computing robustness virtue of GAs and the huge capacity, fast computing virtue of parallel FEM, overcome the long time consuming disadvantage of GAs, acquire high efficiency and accurateness. (2) The program is developed on PC cluster, it provided a economical method that can attain high efficiency by small investment. (3) The computational tests on the solution method in this work revealed that the parallel algorithm is right, and the results are robust. It can solve the big geo-engineering back analysis in practices.

6. ACKNOWLEDGMENT

The writer gratefully acknowledges the financial support provided by the Henan Polytechnical Doctor Foundation (648508) for this research project.

7. REFERENCES

[1] Zhang Youliang, Feng Xiating, "Parallel Computing Method in Geotechnical Engineering", Proceeding of the 2001 IRSM International Symposium, 11-14.09.2001, BEIJING ,CHINA, pp.365~368.

[2] O.T. Bruhns, D.K. Anding, "On the simultaneous estimation of model parameters used in constitutive laws for inelastic material behavior", International Journal of Plasticity, Vol.15, No.12, December 1999, pp.1311~1340.

[3] Papadrakakis M, Tsompanakis Y, and Hinton E, "Advanced solution methods in topology optimization and shape sensitivity analysis", Journal of Engineering Computting, Vol.13, No.5, May 1996, pp.57~90.

[4] The Beowulf project. http://www.beowulf.org.

[5] Sterling T L, Salmon J, Becker D J., "How to build a Beowulf", Cambridge: MIT Press; 1999.

[6] S N Patnaik, A S Gendy, D A Hopkins, "Design optimization of large structural systems with substructuring in a parallel computational environment", Computing Systems in Engineering, Vol.5, No.4-6, 1994, pp. 425~440.

[7] G Hipper, D Tavangarian, "Advanced workstation cluster architectures for parallel computing", Journal of Systems Architecture, Vol.6, No.4, April 2001,pp. 120~126.

[8] Duc T, Nguyen et al, "Parallel finite element domain decomposition for structural/acoustic analysis", Journal of Computational and Applied Mechanics, Vol.4, No.2, pp.189~201.

[9] Shun Doi,Shoichi Koyama, "A parallel computation technique for finite element methods", Systems Computers Controls, Vol.13, June 1982, pp. 7~84.

[10] Charbel Farhat, Edward Wilison, "A new finite element concurrent computer program architecture", J. for Nummer.Meth. in Eng., Vol.24, 1987, pp. 1771-1792

[11] Charbel Farhat, "A Simple an Efficient Automatic FEM Domain Decomposer", Comp. & Struct., Vol.28, No.5,May 1988, pp. 579~602.

[12] James G.Malone, "Automated Mesh Decomposition and Concurrent Finite Element Analysis for Hypercube Mutiprocessor Computers", Computer Methods in Applied Mechanics and Engineering, Vol.70, 1988,pp. 27~58.

[13] Sonzogni V.E, Yommi A.M, "A parallel finite element program on a beowulf cluster", Advances in Engineering Software, Vol.33, No.7, July 2002, pp.427~443.

[14] Petr Krysl, Zden Bittnar, "Parallel explicit Finite element solid dynamics with domain decomposition and message passing: dual partitioning scalability", Computers and Structures, Vol.9, No.4, April 2001, pp.345~360.

[15] Hsin-chen, Numerical linear algebra algorithms on the Cadar system. New York: AMD-Vol86, 1987.

[16] Gullerud A.S, Dodds R.H, "MPI-based implementation of a PCG solver using an EBE architecture and preconditioner for implicit 3-D finite element analysis", Computer and Structure, Vol.79, No.4, April 2001, pp.553~75.

[17] Saad Y. Iterative methods for sparse linear systems. Boston: PWS. 1996.

[18] RU Z L, Feng X T, and Zhang Y L. "Parallel finite element method analysis of bolted rock mass in underground engineering", Chinese Journal of Rock Mechanics and Engineering, Vol.24, No.1, January 2005, pp.13~17.

[19] Holland J. Adaptation in natural and artificial systems, Cambridge: MIT University of Michigan Press, 1975.

[20] Back T, Hammel U, Handbook of Evolutionary Computation. New York: Oxford University Press, 1997

[21] Thomas M. Warschko, Joachim M., "ParaStation: Efficient parallel computing by clustering workstations: Design and evaluation", Journal of Systems Architecture, Vol.44, No.3-4, April 1997, pp.207~226.

Studies of Stock Leverage Effect Model Based on Parallel Algorithm

Yu LIANG, Xi-nan ZHAO
School of Business Administration, Northeastern University
Shenyang, Liaoning, China
Email: liangyu_0310345@163.com
And
Quan-bo GE
Dept of Electrical Automation, Shanghai Maritime University,
Shanghai, China
Email: qqbt@hziee.edu.cn

ABSTRACT

Parallel algorithm was introduced into the estimation and forecasting of stock return model. In the stock return model, discrete event risk was described by random jump-diffusion process and the leverage effect was described by asymmetric Autoregression Conditional Heteroscedasticity model. The samples were classified into five groups according to the existing of day-of-the-week effect. The estimation and the simulation were processed by parallel computing in five groups of samples. The empirical study on index of Shenzhen security market shows that parallel algorithm can not only increase the computing speed largely but also make the accuracy.

Keywords: Jump-diffusion process, Leverage Effect, Parallel Algorithm, Simulated annealing, Stock Market.

1. INTRODUCTION

"Leverage effect" of stock return was introduced by Black in 1976 and it refer to the phenomenon that bad news or negative shocks have a larger impact on volatility than good news or positive shocks [1]. To account for the possible leverage effect, a series of asymmetric effect Autoregressive Conditional Heteroscedasticity models were developed. Exponential Generalized Autoregressive Conditional Heteroscedasticity model (EGARCH) was proposed by Nelson in 1991. Glosten, Jagannathan & Runkel (1993) studied the monthly return of New York Stock Exchange from April, 1951 to December, 1989 and drew the conclusion that negative shocks could increase the variance while positive shocks could decrease the variance a little [2]. This model is called GJR model. Quadratic Generalized Autoregressive Conditional Heteroscedasticity model (QGARCH) was proposed by Engle and Ng (1993). They compared several asymmetric effect ARCH models and drew the conclusion that GJR could describe the leverage effect most properly [3]. All these models have shown that leverage effects are commonly present in the world's stock markets' volatility [4]. It's the same to Chinese stock market. References [5,6,7] have found that there were asymmetric effects in Chinese stock markets.

However research on the Chinese stock markets is very limited and there are two respects of problems:

The first one is there aren't studies that consider discrete event risk while using asymmetric effect ARCH models. Reference [8] referred that huge events have great effect on volatility of Chinese stock markets. When building stock return models, it's very necessary to consider discrete event risk caused by huge event. Otherwise the inaccurate return models will cause incorrect risk management decision.

The second on is most of the existing algorithm use part of the sample to train the model's parameters and then use the rest sample to validate the model. This kind of algorithm is effective when the number of data is small. When we face a large amount of data, the computing speed of the existing algorithm will become very slow. This phenomenon is becoming more and more obvious according to the rapid development of Chinese stock market.

Based on these problems, this paper will give the following solutions:

Firstly, use random jump-diffusion process to describe discrete event risk as well as use GJR model to account for the possible leverage effect. The return series can be described by a Jump-GJR process whose parameters can be estimated by simulated annealing algorithm. By simulation method, we can simply obtain the distribution of intending return.

Secondly, introduce parallel algorithm to increase the speed of computing. Because Chinese stock market is open from Monday to Friday, and there are day-of-the-week effects which means the average return or volatility of someday in one week is obviously different from the other day in the same week. Reference [9] found that there are "Negative Tuesday Effect" and "Positive Friday Effect" in Shanghai and Shenzhen stock markets. According to this phenomenon, we first classify the samples from Monday to Friday and get five groups of data. Then, process each group of data by parallel algorithm and get five groups of parameters of Jump-GJR models.

Finally, integrate the result of five group data and draw an ultimate conclusion. This method can not only increase the computing speed but also lighten the burthen of processor.

2. STOCK RETURN MODELS

2.1 Jump-diffusion series

Merton (1976) introduced jump-diffusion process to Black-Scholes' pricing formula of option [10]. Then a series of pricing models based on jump-diffusion process were developed. This paper will apply jump-diffusion process to stock return series. The basic return model is written as in (1).

$$\phi_i = \mu + \omega_i + \gamma_i \mathbf{1}_{\{q(i)=1\}}, i = 1,2,... \quad (1)$$

In Eq. (1), ϕ_i is the return of a stock in period i, constant μ is the mean and ω_i is the continuous volatility variance, $\varpi_i \sim N(0,\sigma_i^2)$. γ_i is the discrete volatility caused by discrete event and $E\gamma_i = \theta$, $D\gamma_i = \sigma_\gamma^2$, $\gamma_i \sim N(\theta,\sigma_\gamma^2)$.

$q(i)$ is the jump times in period i. Assuming that one day is one period and $q(i)$ follow the Bernoull's distribution, the jump times in one period is 1 or 0. $\mathbf{1}_{\{\cdot\}}$ is function of indication attribute which takes a value of 1 when $\{\cdot\}$ come into existence otherwise value of 0. $\mathbf{1}_{\{\cdot\}}$ Taking a value of 1 means there is an event that makes the return series jump. Obviously the possibility of jump in one period is λ (the intensity of $q(i)$). Eq. (1) is a two-gene model, which include both continuous volatility gene ω_i and discontinuous volatility gene γ_i. It account for Chinese stock return series better than one-gene model.

2.2 Asymmetric return model

Limited by space, this paper chooses GJR-GARCH (1,1) to account for the possible leverage effect. Based on the GARCH (1,1) model, the conditional variance in the GJR-GARCH (1,1) model is expressed as in (2).

$$\sigma_t^2 = a + be_{t-1}^2 + c\sigma_{t-1}^2 + de_{t-1}^2\xi_{t-1t} \tag{2}$$

In Eq. (2), ξ_{t-1} takes a value of 1 when $e_{t-1} < 0$ and value of 0 when $e_{t-1} > 0$. $e_{t-1} = 0$ is a threshold such that shocks greater than the threshold ($e_{t-1} > 0$) have different effects than shocks below the threshold ($e_{t-1} < 0$). When $d > 0$, negative shocks will have a larger impact on σ_t^2 than positive shocks.

Eq. (1) and Eq. (2) compose a mixed-model (called Jump-GJR) that can express both leverage effects and discrete event risk in stock return series.

2.3 Simulation of model parameters

According to (1), ϕ_i is normally with $E(\phi_i \mid q(i)) = \mu + q(i)\theta$, $D(\phi_i \mid q(i)) = \sigma_i^2 + \quad q(i)\sigma_\gamma^2$. Using Maximum Likelihood Function to estimation parameters in the Jump-GJR model, we can get ϕ's probability density function by full probability formula as in (3).

$$f_\phi(x) = \lambda\left[2\pi(\sigma^2+\sigma_\gamma^2)\right]^{-1/2}\exp\left[-2^{-1}(x-\mu-\theta)^2(\sigma^2+\sigma_\gamma^2)^{-1}\right] + (1-\lambda)\left(2\pi\sigma^2\right)^{-1/2}\exp\left[-2^{-1}(x-\mu)^2\sigma^{-2}\right] \tag{3}$$

The logarithm likelihood function of ϕ is expressed as in (4).

$$L_x = \sum_{i=1}^{T}\ln\left\{\begin{array}{l}\lambda\left[2\pi(\sigma^2+\sigma_\gamma^2)^{-1/2}\right]\exp\left[-2^{-1}(\phi_i-\mu-\theta)^2(\sigma^2+\sigma_\gamma^2)^{-1}\right]+\\ (1-\lambda)\left(2\pi\sigma^2\right)^{-1/2}\exp\left[-2^{-1/2}(\phi_i-\mu)^2\sigma^{-2}\right]\end{array}\right\} \tag{4}$$

In Eq. (4), T is the observation time for the past stock return, σ^2 is determined by Eq. (2). The parameters that should be estimated can be defined as vector $\mathbf{x} = (\mu, a, b, c, d, \lambda, \theta, \sigma_\gamma^2)^{\mathrm{T}}$. Maximizing L can get the estimation of every parameter, so it's an optimization problem. In order to get the global optimal solution, heuristic algorithm can be used such as genetic algorithm, simulated annealing algorithm and so on. This paper adopt simulated annealing algorithm.

Simulated annealing algorithm is the spread of local search algorithm. In theory, simulated annealing can find the global optimal solution that can maximize the function value by promising a proper probability. In this study, maximizing L is equivalent to minimizing $f(\mathbf{x}) = -L(\mathbf{x})$ and the steps are expressed as follows:

Firstly, make sure the configuration and range of the solution. At the same time, give an original temperature t_0. Secondly, give birth to a stochastic solution $\mathbf{x}_k$ under t_k. When $f(\mathbf{x}_k) < f(\mathbf{x}_{k-1})$, $f(\mathbf{x}_k)$ will be accepted as a new solution and when $f(\mathbf{x}_k) \geq f(\mathbf{x}_{k-1})$, whether $f(\mathbf{x}_k)$ can be accepted or not will depend on Metropolis principle. That is to say, the moment r_k is a stochastic number belonging to uniform attribution and $r_k < \exp\left\{-\left[f(\mathbf{x}_k) - f(\mathbf{x}_{k-1})\right]/t_k\right\}$, $f(\mathbf{x}_k)$ will be accepted. Otherwise, $f(\mathbf{x}_k)$ will not be accepted and we will repeat the above process L_k times. Thirdly, let $t_{k+1} = d(t_k)$ and repeat step 2. $d(t_k)$ is a decreasing function of t_k, named decreasing temperature function.

Through the above 3 steps, we can get the parameters that can minimize $-L$. By using Metropolis principle, simulated annealing can find the global optimal solution at 100% level.

3. PARALLEL ALGORITHM

Most existing algorithms are used to analyzing the stock return data directly. As time goes on, the stock data may accumulate to a large amount and the computing speed may become lower and lower. So, we introduce parallel algorithm to increase the computing efficiency.

3.1 Classifying the data

We use the samples period from 1996.12.16 to 2004.6.30. There are altogether 1805 data as shown in Fig. 1.

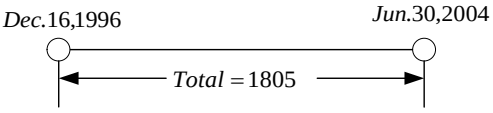


Fig. 1. Total data of Shenzhen Component Index daily return rate

According to the existing of day-of-the-week effects in Chinese stock market [11], we classify the 1805 data into five groups according to Monday, Tuesday, Wednesday, Thursday and Friday. Thus, we can get five new periods of data shown as Fig. 2.

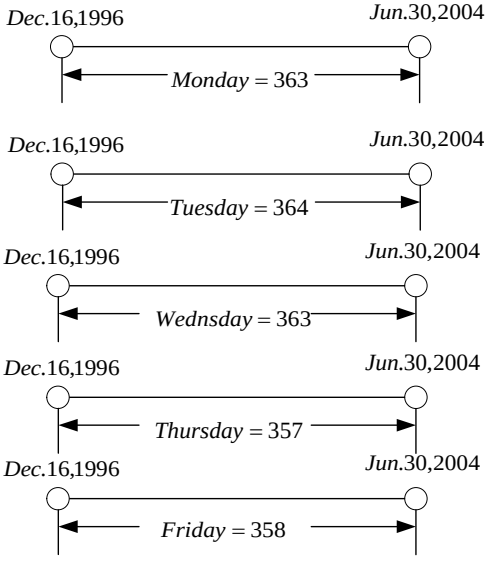


Fig. 2. Five group data of Shenzhen Component Index daily return rate

3.2 Parallel computing

After classifying the data, we distribute each group of data to one computer that belong to one multiprocessor system. Then setup the same program (according to Section II) on each computer. Thus, the whole data can be processed by five computers at the same time. By parallel computing, the computing speed will increase evidently.

4. EMPIRICAL RESULTS OF STOCK INDEX

The data set used in this study covers daily observations from 1996.12.16 to 2004.6.30 for the Shenzhen Component index. There are 1805 return data used to estimate parameters. Each stock index return is calculated as in (5).

$$R_t = \ln(P_t) - \ln(P_{t-1}) \tag{5}$$

In Eq. (5), R_t is index return and P_t is the price of the index. The reason why we didn't use the data before 1996.12.16 is that Chinese stock market began to practice price limits from then on. Under the price limits, the everyday volatility of each stock is limited within 10% and this will affect the parameters in the jump-GJR model. The estimation results of Jump-GJR model for Shenzhen stock market are in Table I.

Table 1. Estimation Results of Jump-GJR model

Para.	Mon.	Tues.	Wed.	Thur.	Fri.
$\hat{\mu}$	6.48e-4	2.36e-3	-9.78e-4	2.17e-3	6.48e-4
$\hat{a}$	6.91e-4	1.46e-3	5.28e-4	1.38e-3	6.91e-4
$\hat{b}$	4.34e-2	-1.22e-2	6.98e-2	-3.30e-2	4.34e-2
$\hat{c}$	3.54e-1	1.50e-1	2.99e-1	1.66e-1	3.54e-1
$\hat{\lambda}$	1.26e-1	1.63e-1	1.97e-1	2.34e-1	1.26e-1
$\hat{\theta}$	-2.69e-2	5.62e-2	-5.67e-2	-2.55e-2	-2.69e-2
$\hat{\sigma}_\gamma^2$	6.19e-2	5.98e-2	4.34e-2	3.31e-2	6.19e-2
$\hat{d}$	4.36e-1	6.87e-1	4.54e-1	1.23e-1	4.36e-1

As shown in Table 1, leverage effect is present in Shenzhen stock market because $\hat{d} > 0$ exists in each group. That means negative shocks do have a larger impact on the volatility of stock index than positive shocks. Moreover, jump intensity $\hat{\lambda} \in [0.126, 0.234]$ means Shenzhen stock index may jump once in 4-8 days. So, the value of parameters evaluated by history data show that it's rational and necessary to consider discrete event risk as well as leverage effect of stock return.

In order to test the forecasting precision of Jump-GJR model by parallel computing algorithm, we forecast the daily volatility interval of Shenzhen index series from 2004.7.1 to 2006.4.21. Using interval estimating at 95% confidence degree, we can get the forecasting volatility interval of Shenzhen index series. There are altogether 415 index data from 2004.7.1 to 2006.4.21 (including 79 Monday, 81 Tuesday, 84 Wednesday, 86 Thursday and 85 Friday). Fig. 3-Fig. 7 displays the relationship between the forecasting volatility interval and the real index return data.

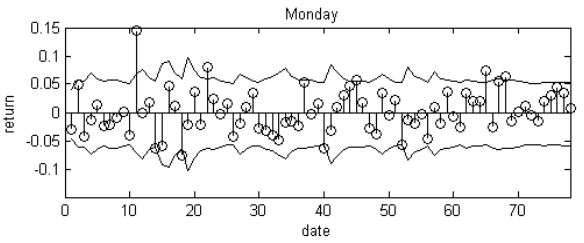


Fig. 3. Interval estimation of Shenzhen Component Index (Monday)

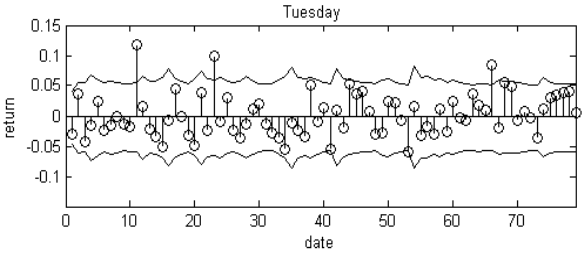


Fig. 4. Interval estimation of Shenzhen Component Index (Tuesday)

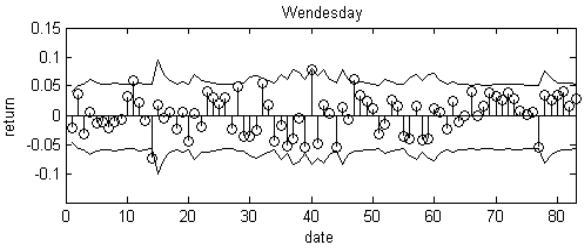


Fig. 5. Interval estimation of Shenzhen Component Index (Wednesday)

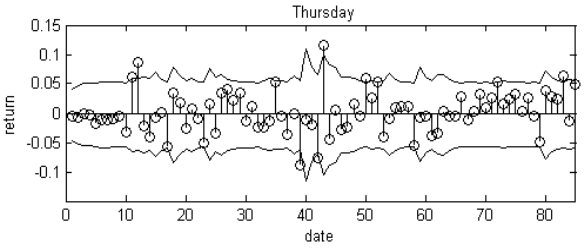


Fig. 6. Interval estimation of Shenzhen Component Index (Thursday)

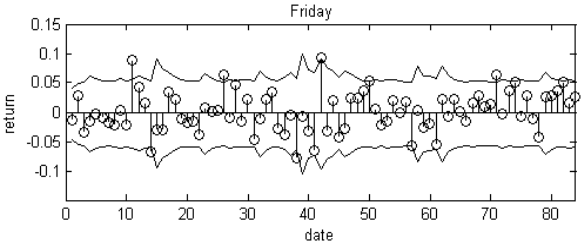


Fig. 7. Interval estimation of Shenzhen Component Index (Friday)

Table 2. Forecasting Precision of Jump-GJR Model by Parallel Computing

Number of real data included by each forecasting curve		Forecasting precision of each group of data	
Mon.	75	Mon.	95.00%
Tues.	78	Tues.	96.30%
Wed.	81	Wed.	96.43%
Thur.	81	Thur.	95.35%
Fri.	81	Fri.	95.29%

The curve in every figure is the daily volatility interval of index return estimated by GJR-GARCH model, which consider not only the leverage effect but also discrete event risk. In the middle of the curves, there are real index return series from 2004.7.1 to 2006.4.21. Studying on how many real return data that included by the forecasting curves, as in Table 2, we can see the forecasting precision of each curve in Shenzhen market can reach 95%. If we do not separate the total data into five groups and do not use parallel computing method to process the data, the forecasting precision of JUMP-GJR model in Shenzhen market will be 94% which is lower than that of parallel computing. Moreover, under the same condition, the parallel computing time including both estimation and forecasting is 13 minutes for each group while 30 minutes if parallel algorithm is not used. So, applying parallel algorithm to the stock return model can make the process more efficient as well as accurate.

5. CONCLUSION

This paper applied both jump-diffusion process and parallel algorithm to asymmetric effect ARCH models. The empirical study on index of Shenzhen security markets shows applying parallel algorithm to the stock return model can make the process more efficient as well as accurate. This method that applying parallel algorithm to stock return models will be used widely as Chinese stock market develops continuously.

6. REFERENCES

[1] F. Black, “Studies of stock market volatility changes”, Proceedings of the American Statistical Association, Business and Economic Statistics Section, 1976, 177-181.

[2] B. Chen, “Empirical study on Chinese stock market’s leverage effect and intra-market transmission of conditional volatility”, Master dissertation, Univ. Xiamen, China, 2001.

[3] L. R. Glosten，R. Jagannathan, et al，“On the relation between the expected value and the volatility of the nominal excess return on stocks”, Journal of Finance, 1993, Vol.48: 1779-1801.

[4] Ng V.， R. Engle, et al, “A multi-dynamic-factor model for stock returns”, Journal of Econometrics, 1992，Vol.52: 245-266.

[5] L.L. Li, “Study on Chinese stock market leverage effect”, Securities Business Newspaper, 2002, Vol.10: 10-14.

[6] X.Q. He, Q.Y. Sun, “The leverage effect and equilibrium of risk return in Chinese stock market”, NANFANGJINGJI, 2003,17(9): 62-65.

[7] L.F. Meng, S.Y. Zhang, et al, “Bayesian analysis of stochastic volatility model with leverage effect and its application”, Systems Engineering, 2004, 22(3): 50-53.

[8] L.B. Zhang, Y. Liang, et al, “Study on value at risk based on event risk”, China management science, 2005,13(2): 113-117.

[9] H.B. Wang, X. Yang, “Day-of-the-week effects of tradingvolume in Chinese stock market”, Journal of Tsinghua University, 2004,44(12): 1615-1617.

[10] R.C. Merton, “Option Pricing When Underlying Stock Returns Are Discontinuous,” Journal of Economics, 1976, No.3: 123-141.

[11] D.H. Jin, Study on Chinese stock market volatility and control, NJ: Shanghai financial university Press, 2003, 239–245.

A Parallel Processing Method of Divide-and-Conquer and a Highly Efficient Parallel Sorting Algorithm

Minghe Huang, Cuixiang Zhong, Liping Dai, Gang Lei
Software Engineering College, Jiangxi Normal University
Nanchang, Jiangxi, 330022, China
Email: huangmh0093@sina.com

ABSTRACT

With the rapid development of parallel processing, the research of parallel processing methods of general algorithm design techniques such as divide-and-conquer becomes more and more significant. This paper discusses a parallel processing method of divide-and-conquer and its applications to sorting operations on cubes. It derives a highly efficient parallel sorting algorithm that can achieve asymptotically suboptimal $O(\log^2 n) = O(\log^2 p)$ running time when $n = p$. This new parallel sorting algorithm is presently one of the fastest practical deterministic sorting algorithms available.

Keywords: parallel processing, general algorithm design techniques, divide-and-conquer, cubes, the parallel sorting algorithm.

1. INTRODUCTION

With the rapid development of parallel processing, more and more problems can be solved on parallel computers. But we're still lacking in parallel processing methods of general algorithm design techniques. Mature and general parallel algorithm design techniques have not been formulated yet. The algorithms designed by general sequential algorithm design techniques such as divide-and-conquer aren't suitable to general computers because their performance is very low on these systems. That has seriously hindered the solution of many practical problems [1,3,4,7]. Hence, according to sequential algorithm design techniques and the rule of parallel processing, the authors have studied deeply the parallel processing methods of algorithm design techniques such as divide-and-conquer, and have summarized a useful parallel processing method for divide-and-conquer. In order to obtain a highly efficient parallel sorting algorithm, the authors have also applied this parallel processing method to sorting operations.

This paper mainly introduces a parallel sorting algorithm. The general problem of sorting on a hypercube is as follows: Given n records distributed evenly among the $p = 2^q$ processors of a q-cube (with each processor holding n / p records) rearrange the records so that the key values are in the same order as the processor node labels. Ideally, we expect an optimal running time of $T = O((n\log n)/p)$ and linear speed-up. Yet currently we cannot achieve this optimal running time for all values of n and p. For the special case of $n=p$, i.e., each processor holding a single record before and after sorting, the most practical hypercube sorting algorithms are based on Batcher's odd-even merge. Batcher's sorting algorithms achieve asymptotically suboptimal $O(\log^2 n) = O(\log^2 p)$ running time. The new parallel sorting algorithm introduced here is based on Batcher's bitonic sorting method, and can also achieve asymptotically suboptimal $O(\log^2 n) = O(\log^2 p)$ running time. The best known deterministic sorting algorithm requires $O(\log p \log\log p)$ or $O(\log p(\log\log p)^2)$ running time for the case $p=n$ in the worst case, depending on assumption. But it is not competitive for $p \le 2^{20}$, so the new parallel sorting algorithm introduced here is presently the fastest practical deterministic sorting algorithm available [2,3,5,6].

2. THE PARALLEL PROCESSING METHOD OF DIVIDE-AND-CONQUER

All printed divide-and-conquer is an important general algorithm design technique. Its main idea is as follows [1,7]: For a problem of size n, we can divide it into k subsets in some way based on its own properties, $1 \le k \le n$, so that there are m subproblems of the same properties. The m subproblems are solved, and then the solutions obtained for these subproblems are combined to get a solution to the original problem in some way. If a subproblem is still too big, it could be divided into smaller subproblems by means of divide-and-conquer until these smaller subproblems are small enough to solve. So divide-and-conquer can be expressed by recursive process. The parallel processing of divide-and-conquer considers the problem solving on the parallel computers instead of the general serial computers to make some operations run simultaneously during the solving process.

Assuming for simplicity that n is a power of 2, i.e., $n = 2^d$ where d is an integer, we use ALGORITHM (0,n-1) to describe the parallel processing of divide-and-conquer. The input data are stored in a one-dimensional array T, and the elements of this array T are changed during the problem solving. At the end of algorithm execution, the content in T is the output result. The following paragraph is the general description of ALGORITHM [7].

```
Procedure ALGORITHM(l, l+m-1)
Begin
    ALGORITHM(l, l+m/2-1);
    ALGORITHM(l+m/2, l+m-1);
    For each i:l≤i<l+m/2 pardo
        (T[i],T[i+m/2])←OPER_i(T[i],T[i+m/2])
End;
```

We can also employ loop to take the place of recursion. Let $bit_i(j)$ denote the $(i+1)th$ digit of the binary representation of integer j from right to left. So the ALGORITHM could be described as follows [7].

```
Procedure ALGORITHM1(0,n-l)
Begin
  for i←0 to d-1 do
    for each j:0≤j<n pardo
      if bit_i(j)=0 then
        (T[j],T[j+2^i])←OPER_i(T[j],T[j+2^i])
End;
```

In fact, we can implement the above parallel processing algorithm of divide-and-conquer on cube structure. Assuming that the cube structure consists of *n* nodes, which we number from 0 to *n*-1, each node *i* stores an element *T[i]*. The edge between the node storing *T[i]* and the node storing *T[j+2^i]* expresses an operation on these two elements. So the cube structure is a *d*-dimensional cube, as shown in Figure 1. All the edges in one direction is called *dimension*, and all edges of the cube are divided into *d* dimensions, each dimension has *n/2* edges. During the realization of ALGORITHM, each step in the loop uses parallel all the edges of some dimension, and the value of the variable *i* is the dimension number.

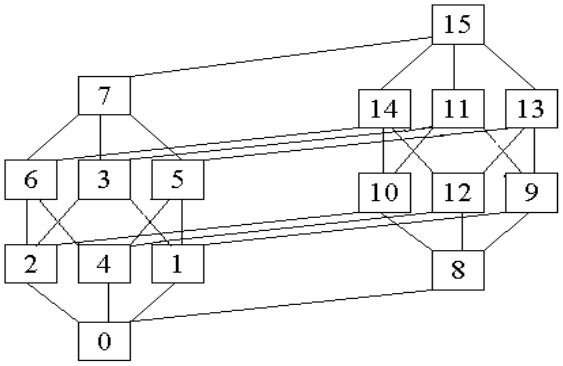


Figure. 1. Cube structure of ALGORITHM1 (0,15)

The operation order in ALGORITHM1 is from low dimension to high dimension. This kind of parallel algorithms of divide-and-conquer is called ASCEND, while another kind of parallel algorithms of divide-and-conquer is called DESCEND whose operation order is from high dimension to low dimension, i.e., changing *for i←0 to d-1* in ALGORITHM1 into *for i←d-1 to 0*.

3. A HIGHLY EFFICIENT PARALLEL SORTING ALGORITHM

3.1 Algorithm description

As an important application of the parallel processing of divide-and-conquer, we put forward a parallel sorting algorithm on cube structure. The following is the detailed description of the sorting algorithm.

Using the parallel divide-and-conquer algorithm of descending order, it is easy to sort a bitonic sequence to a monotonic sequence [2,3]. A bitonic sequence is a sequence $Z = (z_1, z_2, \ldots z_n)$ such that at least one of the following conditions holds: (1) $z_1 \geq z_2 \geq \ldots \geq z_i \geq z_{i+1} \leq \ldots \leq z_n$ for some integer i, $1 \leq i \leq n$. (2) Z could be transformed to satisfy the former condition by circular shifting. The following algorithm can easily transform a bitonic sequence to an ascending sequence, especially when Z satisfies $z_1 \geq z_2 \geq \ldots z_{n/2} \leq z_{n/2+1} \leq \ldots \leq z_n$ where $n = 2^d$.

```
Procedure ALGORITHM2 (0,n-l)
Begin
for i←d-1 to 0 do
foreach j:0≤j<n pardo
if bit_i(j)=0 then
(T[j],T[j+2^i])←(min(T[j],T[j+2^i]),max(T[j],T[j+2^i]))
End {of procedure}
```

Obviously the running time of this algorithm is $O(\log n)$.

If the operation step in the above algorithm is changed into: $(T[j],T[j+2^i]) \leftarrow (\max(T[j],T[j+2^i]), \min(T[j],T[j+2^i]))$, then we can obtain another algorithm called ALGORITHM3(0, *n*-1), which sorts *Z* to a monotonic descending sequence.

Based on these algorithms, we can design a new parallel algorithm that can sort any 2^d sequence to a monotonic sequence. The basic idea of the new algorithm is: Assuming that the array *T(n)* keeping the sequence is put in a cube structure in a way described in the above context, then *d* steps are needed to implement the sorting , the *kth* ($1 \leq k \leq d$) step divides the group with *n* nodes into 2^{d-k} subgroups, and ALGORITHM2 or ALGORITHM3 are used to process these subsequences in these subgroups. The following paragraph is the description of this algorithm.

```
Procedure Parallel_Sort(0,n-l)
Begin
for k←1 to d do
  for i←k-1 to 0 do
    for each j:0≤j<n pardo
      if bit_i(j)=0 then
        if 0≤j mod 2^(k+1)≤2^k-1 then
(T[j],T[j+2^i])←(max(T[j],T[j+2^i]),min(T[j],T[j+2^i]))
        else
(T[j],T[j+2^i])←(min(T[j],T[j+2^i]),max(T[j],T[j+2^i]))
End {of procedure}
```

3.2 Algorithm analysis

Analyzing the above algorithm, we find that its operations are divided into $d = (\log n)$ steps, and the *kth* ($1 \leq k \leq d$) step adopts ALGORITHM2 (l,l+2^k-1) or ALGORITHM3(l,l+2^k-1) to parallel process 2^{d-k} subsequences. Based on the analysis of ALGORITHM2, the running time of these algorithms processing sequence of 2^k elements is $O(\log 2^k)$ or $O(k)$. So the running time of all d steps is:

$$1 + 2 + \cdots + d = d(d+1)/2 = \log n(1 + \log n)/2 = O((\log n)^2)$$

Compared with the existing parallel sorting algorithms, this algorithm's running time is much lower, so it is a highly efficient algorithm.

4. CONCLUSION

Although parallel processing develops rapidly, mature and general parallel algorithm design techniques have not been formulated yet. Hence, the study of parallel processing methods of general algorithm design techniques is significant. This paper has discussed the parallel processing method of divide-and-conquer, and has summarized a useful parallel processing method for divide-and-conquer. The authors have also applied this parallel processing method to sorting operations to obtain a highly efficient parallel sorting algorithm, which is based on Batcher's bitonic sorting method

and can achieve asymptotically suboptimal $O(\log^2 n) = O(\log^2 p)$ running time. Although the best deterministic sorting algorithm for the case $p = n$ is known as having $O(\log p \log\log p)$ or $O(\log p(\log\log p)^2)$ running time in the worst case, it is not competitive for $p \le 2^{20}$, so the new parallel sorting algorithm introduced here is presently one of the fastest practical deterministic sorting algorithm available. The parallel processing method of divide-and-conquer and its applications to sorting operations on cubes introduced here will certainly inspire the study of processing method of divide-and-conquer and its applications on other architectures such as linear array, meshes.

5. REFERENCES

[1] B. Wilkinson, M. Allen, Parallel Programming—Techniques and Applications Using Networked Workstations and Parallel Computers, Higher Education Press, Beijing, 2002.

[2] K. Batcher, "Sorting Networks and Their Applications", Proc.AFIPS Spring Joint Computer Conf., 1968,Vol.32, pp. 307-314.

[3] B. Parhami, Introduction to Parallel Processing—Algorithms and Architectures, Plenum Press, New York, 1999.

[4] G. L. Chen, Parallel Algorithm Practice, Higher Education Press, Beijing, 2004.

[5] R. Cypher, G. Plaxton, "Deterministic Sorting in Nearly Logarithmic Time on the Hypercube and related Computers", Proc.22nd ACM Symp.Theory of Computing, 1990, pp.193-203.

[6] J. Reif, L.Valiant, "A logarithmic Time Sort for Linear size Networks", J.ACM,34(1), PP. 60-76,1987

[7] H.Zhu, Algorithm Design and Analysis, Shanghai Science and Technology Literature Press, Shanghai, 1989 (In Chinese).

Rate Limiting with Network Monitor Approach to Counter DDoS Attacks in Distributed Computing Environments

Ong Ghim Hwee
School of Computing, National University of Singapore
Republic of Singapore
Email: onggh@comp.nus.edu.sg
Koh Wee Kiat
School of Computing, National University of Singapore
Republic of Singapore
Email: kohweeki@comp.nus.edu.sg

ABSTRACT

In the mitigation of DDoS attacks, rate-limiting seems to be one of the viable strategies. On top of that, cooperative security management can help achieve higher effectiveness of DDoS defence mechanisms. Two rate-limiting DDoS mitigation mechanisms, the pushback mechanism and the level-k max-min mechanism are compared in this paper. A network monitor simulation model is also proposed and implemented. Using the survivability and delay metrics defined, simulation results showed that clear distinctions in router congestion levels can be identified during a DDoS attack, showing the potential of such a tool in gauging the effectiveness of DDoS defences, possibly in distributed computing environments.

Keywords: Distributed Denial-of-Service (DDoS) attacks, Rate-limiting Mechanisms, Distributed System Architecture.

1. INTRODUCTION

A Denial-of-Service (DOS) attack does not involve breaking into the target machine, but aims to disrupt the services being offered by a web-server by either consuming the resources of the victim machine or flooding its network, often with a large merciless stream of information. This is often in the form of 'ping' messages. Responding by creating a connection, the target machine will not be able to return the message if a non-existent address had been provided. It will then have to wait till the connection times out. In large amounts, this results in service degradation, and may prevent legitimate users from obtaining services.

While a DOS attack usually involves only a single attacker and results in damage of a smaller scale, Distributed Denial-of-Service (DDoS) attacks have the potential to inflict much more damage to its target machines. Expanding on the idea of DOS attacks, a DDoS attack is launched by a group of compromised hosts (often known as 'zombies'). Such an attack is coordinated to send an extremely large volume of traffic to the target server, usually resulting in the victim server being overloaded and possibly incapacitated.

Various defence mechanisms against DDoS have been proposed and implemented. In general these mechanisms can be generalized and classified according to their approach [1]. Firstly, *intrusion prevention* mechanisms try to stop DDoS attacks from happening. Methods such as the identification of attack packets at source and the reduction of systems vulnerabilities are used. *Intrusion detection* mechanisms try to detect DDoS attacks so that prompt action can be taken. This can be done either by identifying attack signatures or looking out for statistical anomalies in traffic patterns. Lastly, *intrusion mitigation and tolerance* mechanisms accept that DDoS attacks will occur, and tries to contain the adverse effects caused by such attacks. The aim of such mechanisms is to ensure that a certain proportion of legitimate users are still able to obtain required services in the even of an attack.

The remaining discussion will be organized in the following manner: In Section 2, we will review some background of rate-limiting as a defence mechanism, a comparison between two such mechanisms, as well as cover a short introduction to the concept of network monitoring in a distributed environment. We will then consider the implementation of the network monitor simulation model in Section 3. The simulation results will be presented in Section 4. Finally we conclude the discussion in Section 5.

2. BACKGROUND AND RELATED WORK

Overview of Rate-Limiting

Rate-limiting generally refers to the restriction of data flow of one node to another in a network. In this discussion, we consider only server-centric rate-limiting mechanisms. A server-centric approach refers to the technique where defensive actions are initiated by the server under attack. The rate-limiting mechanisms are deployed on routers which are a few hops away from the server, such that attack traffic can be handled more effectively before aggregation.

An interesting problem to consider in such mechanisms is the issue of rate computation. Given that rate-limiting may be employed on multiple routers in the event of an attack, it is important to ensure that the allowed rates assigned to each router be calculated in a fair fashion. It should also be dynamically re-calculated depending on network conditions, so as to ensure that any degradation of services to legitimate users is kept to a minimum.

For the purpose of our discussions, a closer look will be taken at two server-centric rate-limiting mechanisms, pushback and level-k max-min.

The Pushback Mechanism

The pushback mechanism [2] is part of the concept of *Aggregate-based Congestion Control* (ACC), whereby an aggregate is defined as a subset of network traffic with some property that can be identified. The main purpose of the pushback mechanism is to locate aggregates that are responsible for server or network congestion, and initialize action to drop packets belonging to such aggregates at the routers.

Consider the illustration in Fig. 1., where server S is under a DDoS attack. We first consider the situation at router R1. Faced with traffic congestion, traffic from routers R2, R3 and R4 will be selectively dropped if R1 cannot cope with the incoming traffic. In such a case, we observe that traffic from R4 is being treated unfairly; good traffic following down the link R4-R1 is dropped due to attack traffic from the other links.

Before pushback is invoked in this case, the ACC mechanism will first determine the aggregates that are causing the congestion, and neighbouring nodes are categorized into contributing and non-contributing links. A contributing link is one that contributes a high amount of attack traffic. After that, a pushback message is sent via each contributing link, requesting for traffic from the offending aggregates to be rate-limited. Since contributing links do not necessarily provide the same amount of traffic, the rate for which each link is to be limited has to be computed using any reasonably fair algorithm.

Pushback is a recursive mechanism; therefore the process can be continued further upstream in a similar fashion if needed. In the above example, pushback messages can be further propagated to router R5, such that most of the good traffic from R6 can be protected as well.

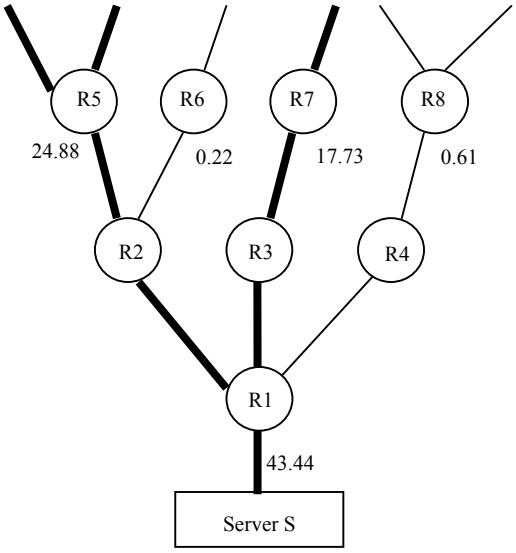


Fig. 1. A DDoS attack taking place. Darkened lines indicate links with attack traffic.

The Level-k Max-Min Mechanism

The key feature of the level-k max-min mechanism [5] involves the installation of throttles at routers that are k-hops away from the server to be protected (hence the name level-k). The main objective of this mechanism is for the server to extend its maximum allocated service bandwidth and no more. In other words, we want the server to be providing the maximum bandwidth it is allowed to service users, even when it may be under attack.

The algorithm defines two parameters that represent the upper and lower bounds of the load limit, L_s and U_s respectively. The mechanism starts off by first assigning a throttle rate r_s to the routers located at level-k. The traffic arrival rate (server load) ρ is then monitored to determine if it meets the criteria of $L_s < \rho < U_s$. The throttle rate is halved if $\rho > U_s$, and a constant factor δ is added if $\rho < L_s$. This loop continues until the traffic arrival rate meets the above-mentioned criteria, or it is determined that there is not significant increase in traffic arrival rate even after the adding of δ (i.e., throttle is already optimal).

We now consider the example shown in Fig. 1.. Let parameters $L_s = 12$, $U_s = 16$ and $\delta = 1$. Before applying the level-k max-min mechanism, we first initialize $r_s = (L_s + U_s)/2 = 10$ and decide on $k = 3$. The initial server load is 43.44, well over the ideal range we desire. When the throttle is first invoked, the traffic rates from the affected nodes will be 10, 0.22, 10 and 0.61 respectively. Note that the throttle does not apply to nodes, which already have traffic rate lower than the throttle rate. The server load is now 20.83, which is larger than the range required. The throttle rate is then halved to 5, giving traffic rates of 5, 0.22, 5 and 0.61. The server load is now 10.83 which is too low, so $\delta = 1$ is added to give a throttle rate of 6. With traffic rates of 6, 0.22, 6 and 0.61, an ideal server load of 12.83 is obtained and the mechanism terminates.

Motivation for Use of Network Monitor

Through the use of a network monitor, it is hoped that information can be collected when throttling is in place, possibly to make a probabilistic evaluation of the nature of the DDoS attack that may be taking place. This is achieved by sending a small number of network monitoring packets to routers in the network and computing certain metrics which will provide an idea of the current traffic conditions at the particular router.

Such an approach may assist network administrators in the evaluation of the existing problem as well as to undertake other defensive actions. More importantly, it can potentially lessen the computational demands placed upon routers in DDoS defence mechanisms.

There are a number of advantages associated with the use of a network monitoring mechanism, which we will discuss at this point. The first advantage relates directly to the two defence mechanisms discussed earlier. Both mechanisms require resources that are under attack to be able to initiate and sustain defensive actions. In the case of level-k max-min, the server under attack has to continually monitor traffic conditions and provide throttling feedback. The pushback mechanism also requires intermediate routers to exchange messages and decide whether to further propagate defensive actions upstream. It may be a little too much to ask for such action from nodes which are already congested due to the attack. Therefore it may be more beneficial and reliable to employ an external network monitor that is able to probe the routers taking part in defensive actions for their effectiveness.

Secondly, the pushback and level-k max-min mechanisms employ the use of aggregated traffic rates. While this is sufficient to ensure that the aggregated traffic rate reaching the server is within its processing limits, it does not ensure optimum performance at the rate-limiting routers. For example, a router that is under rate limiting may be dropping 90% of all traffic but it will be deemed as effective as long as the server receives an optimum volume of traffic. With the use of the network monitor, it is possible to judge the effectiveness of defence actions not only through the traffic rate the server receives, but also the amount of traffic that is sacrificed to achieve that aim.

Last but not the least, the use of a network monitor supports the notion for cooperative management, possibly using distributed system architecture. This will be covered in greater detail in the next segment of our discussion.

Defences for Distributed System Architecture

While the above rate-limiting mechanisms were introduced using a single server example, they can be further extended to serve multiple servers in a networked system. In many senses, a networked system has a distributed architecture, with facilities such as routers, switches and servers inter-connected.

An overview of a networked information system as a distributed architecture was presented by Sung et. al.[4]. They presented a method for cooperative management in a distributed environment in order to ensure a higher level of service survivability. In their model, a networked system can

be divided into several domains of which a particular node within a domain is assigned as a domain manager, which will then be responsible in monitoring traffic at the edge routers and controlling the usage of resources in its domain. The objective of the network monitoring performed is to detect the presence of any abnormal inbound traffic flows. Messages are exchanged among domain managers once an attack is detected such that traffic to the victim node may be cut off.

Such a cooperative structure can also be utilized if rate-limiting is implemented across networks. Given that the routers at k-hops away from a server could easily reside in different network arms, different network monitors can monitor and exchange information on the effectiveness of the rate-limiting mechanisms in action on the routers within their logical domain.

3. SIMUALTION IMPLEMENTATION

For the purpose of our discussions, simulations were carried out using GPSS World™ [3], which is a discrete event simulator well suited for resource management problems. Using this simulator, a simulation model is viewed from the perspective of the lifetime of a data packet.

Choice of Rate-Limiting Mechanism

A comparison was done between the level-k max-min and the pushback mechanisms in order to select one for use in our network monitor simulation model. Simulations were performed based on a network consisting of a single server and a number of users that are a number of hops away from the server [5]. Out of these users we choose a subset of nodes to be attackers with a probability of p.

Simulations were conducted using a network structure resembling that of a 3-nary tree. The network model that is implemented extends to a level k = 5, with 363 router nodes and 729 user nodes. For the level-k max-min mechanism, the server is modeled with the parameters $L_s = 656.1$ and $U_s = 801.9$. The values are computed based on an expected average traffic rate of 1 Kb/s per normal user. A step size $\delta = 0.3$ is used for stable system performance. As for the pushback mechanism, the interval whereby the pushback signal is propagated upstream is set at a value of 2 second

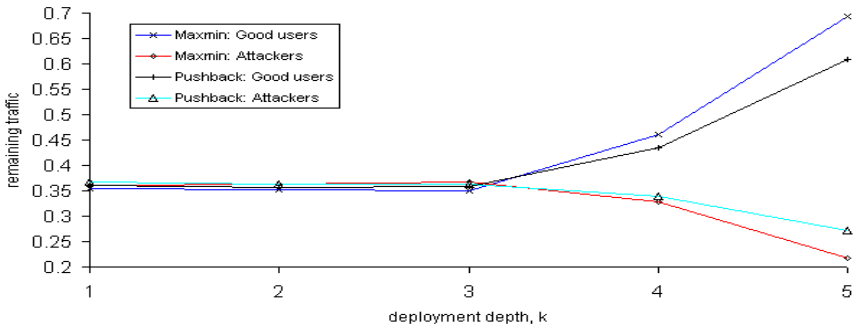


Fig. 2. Results for $r_g = 2$, $r_a = 20$ and $p = 0.2$

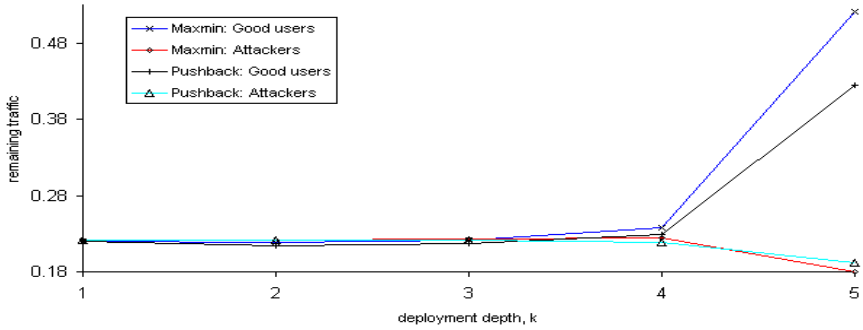


Fig. 3. Results for $r_g = 2$, $r_a = 20$ and $p = 0.4$

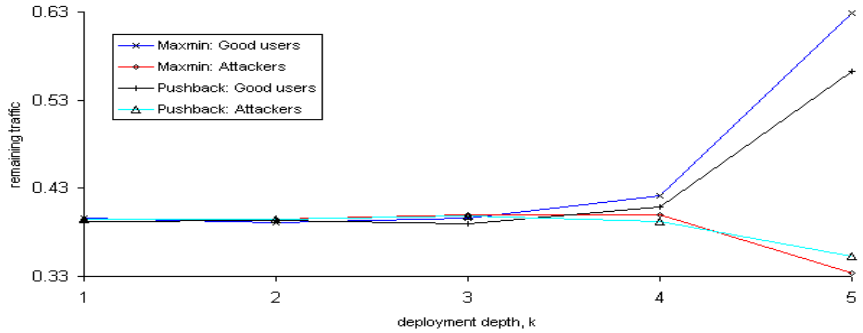


Fig. 4. Results for $r_g = 2$, $r_a = 10$ and $p = 0.4$

The key performance metrics used to evaluate the effectiveness of each mechanism are: (1) the percentage of good

user traffic that makes it to the server and (2) the percentage of attacker traffic that arrives at the server.

The results of the set of simulations where attacker and good user nodes are evenly distributed in the network are shown in Figs. 2, 3 and 4. The parameters r_g and r_a represents the highest data rate possible from a good user and an attacker respectively. We are able to observe that for all the cases, both mechanisms give a similar performance in the percentage of remaining attacker and good user traffic between the levels k = 1 and k = 3. However, as k increases to 5, there is a significantly better level of protection for the level-k max- min mechanism as compared to the pushback mechanism. This is due to the fact that good traffic aggregates to a considerable level near server *S*, making it difficult to differentiate between traffic from good users and attackers. Hence, the max-min mechanism, which initiates defence actions further away from *S* is able to give a better level of protection as compared to the pushback mechanism. Simulations were also carried out for attackers that were distributed unevenly across the network. This is achieved by selecting four disjoint sub-trees from the network, as shown in Fig. 5.. The various configurations for the uneven distributed attacker simulations are as follows, where the attacker concentration increases from configuration 0 to configuration 3.

- Configuration 0: Entire graph
- Configuration 1: Sub-trees 2 , 3 and 4
- Configuration 2: Sub-trees 1 and 4
- Configuration 3: Sub-trees 1 and 2

The percentage of good user traffic that remains for the various configurations under the level-k max-min mechanism is shown in Fig. 6.. The corresponding results for the pushback mechanism are shown in Fig. 7.. One key observation is that for each of the mechanisms, the level of protection generally improves from configuration 0 to 3. This is because a larger percentage of attacker traffic aggregates earlier in the network, a larger portion of attacker traffic can be dropped earlier, and user traffic protected.

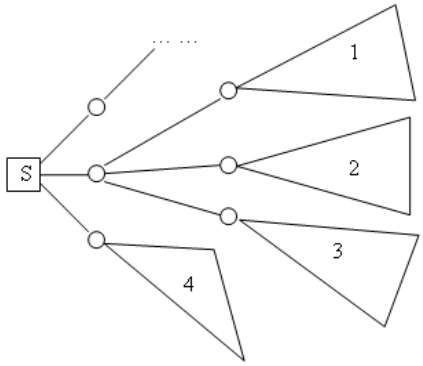


Fig. 5. Sub-trees used for the uneven distributed attacker simulations

In Fig. 8., we present the results for configurations 0 and 3 together for both mechanisms such that a comparison can be made. It can be observed that the level-k max-min mechanism consistently outperforms the pushback mechanism across the four configurations.

Given the more favorable results given by the level-k max-min mechanism, it was selected for the implementation of the network monitor simulation model.

Network Monitor Metrics

For the purpose of our simulations, we define two metrics for the network monitor, namely, the survivability metric and the delay metric. These metrics will help us to evaluate the amount of attack traffic that is present at routers a level-k away from the server.

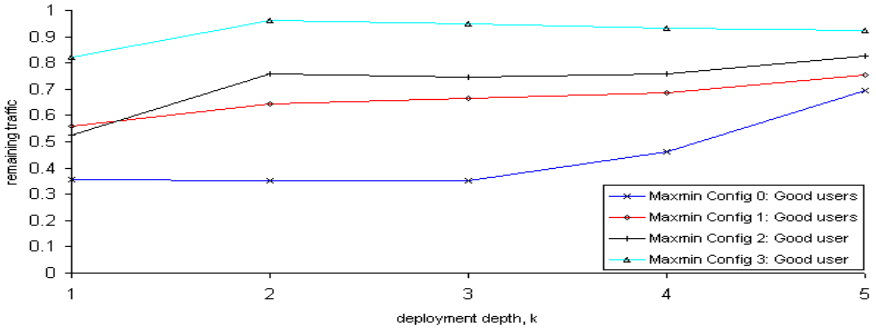


Fig. 6. Percentage of good users under configurations 0 – 3 for the level-k max-min mechanism

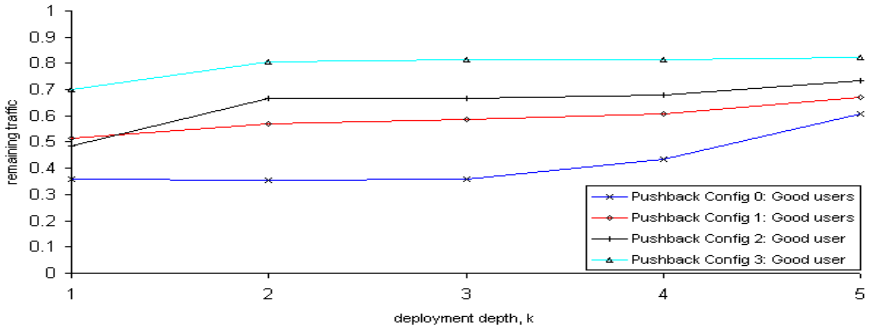


Fig. 7. Percentage of good users under configurations 0 – 3 for the pushback mechanism

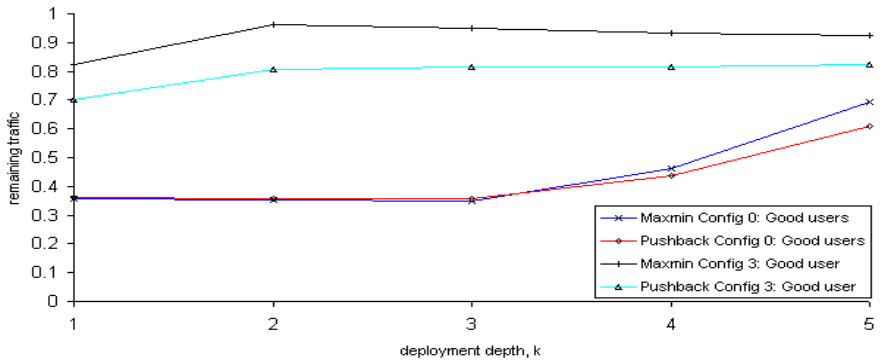


Fig. 8. Comparison between the two mechanisms for configurations 0 and 3

The survivability metric is used as a measure of the percentage of packets that is dropped at a particular router. A total number, N, of marked packets will be sent at time intervals T by the network monitor towards the targeted router. We assume that the network monitor has priority in bandwidth allocation for sending the packet; therefore network delay is kept at a constant minimum. However the packet will not enjoy any priority at the router, which means it will have to join the receive buffer of the router and get dropped when the buffer is full. Upon receiving the packet, the router simply forwards it back to the network monitor. The survivability metric is calculated as follows, where R is the number of packets sent out that are returned:

$$\text{Survivability} = \frac{N}{R} \times 100\%$$

The delay metric measures the average time taken for surviving packets to be replied to the network monitor. We expect the delay to be longer for routers that are more congested as opposed to other routers. Taking d_i to be the delay for received packet i, the delay metric, given in seconds, is computed as follows:

$$\text{Delay} = \left(\sum_{i=1}^{R} d_i\right) / R$$

Network Monitor Simulation Settings

For the purpose of understanding whether a network monitor approach is helpful, we model a series of simulations with the uneven distributed attacker topology described earlier. As a recap, we model four cases based on the parameters $r_g = 2$, $r_a = 20$ and $p = 0.2$, the first case with a random distribution of attackers and subsequent cases with unevenly distributed attackers.

The parameters for the network monitor are set as $N = 10$ and $T = 0.2$ s. This low number of packets is chosen so as not to add to the attack traffic at the router, while the time interval spreads out the monitoring period to compensate for possible cases of flash floods, which are just random bursts of high traffic due to a high number of users accessing the server at the same time.

The network monitor is initiated 5 seconds after the server detects the presence of a DDoS attack and initiates defensive actions. We probe the routers at level k = 3 and try to verify the aggregation of attack traffic through only a subset of routers in the case of unevenly distributed attackers. We label the routers with indexes 1 to 27, where router index 1 refers to the router at level k = 3 at the top of the network tree shown earlier in Fig. 5.. Correspondingly, router index 27 refers to the router at the bottom of the level.

4. SIMUALTION RESULTS AND ANALYSIS

The simulation results for unevenly distributed attackers using configuration 1 is shown in Fig. 9.. Configuration 1 distributes attackers unevenly such that attack traffic is concentrated at the lower half of the network tree. As observed from the results, it is clear that the routers in the lower half of the network suffer from a greatly diminished survivability rate; most likely due to a large number of packets being dropped due to the large volume of attack traffic. It can be noted at the same time that the delay for these congested routers are higher than the rest, presumably due to the need for surviving packets to go through longer queues at affected routers. Hence the network monitor is able to give a good indication of the concentration of attack traffic from a particular arm of the network.

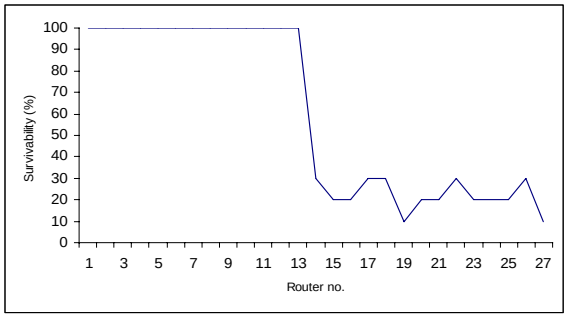


Fig. 9. Survivability metric for configuration 1

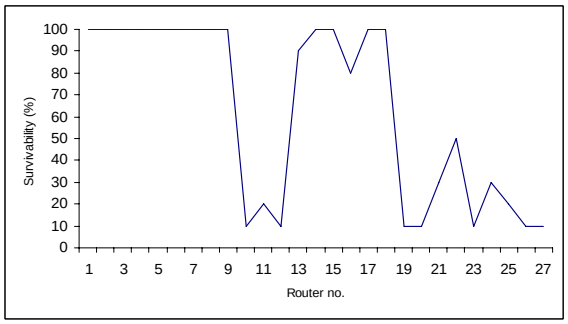


Fig. 10. Survivability metric for configuration 2

Similarly, we are able to identify an uneven attacker distribution when we look at the results of simulation runs for configurations 2 and 3, shown in Fig. 10. and Fig. 11.

respectively. In both cases the survivability of the network monitor packets sees a sharp drop for routers where attack traffic is aggregated. The other routers in the same level maintain their level of performance as no attack traffic is routed through them. It is also interesting to note that in configuration 3, the survivability metric of routers handling attack traffic is somewhat lower to those values in configurations 1 and 2. This is due to the attackers being concentrated in one particular sub tree of the network; hence the aggregated attack traffic volume is extremely large.

Having analyzed the results of previous simulations, it can be seen that a network monitor mechanism can, to a certain degree, provide a third party view on the current state of DDoS attacks by looking at the routers that have undertaken defensive actions. We now consider the simulation for a random distribution of attacker nodes, with results shown below in Fig. 12..

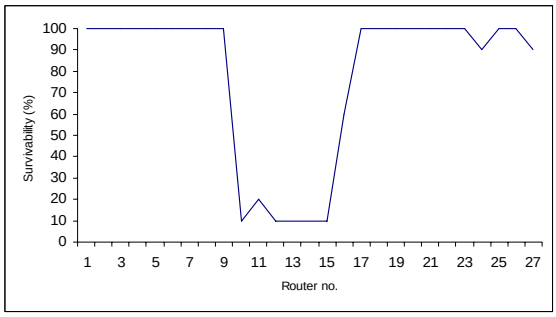


Fig. 11. Survivability metric for configuration 3

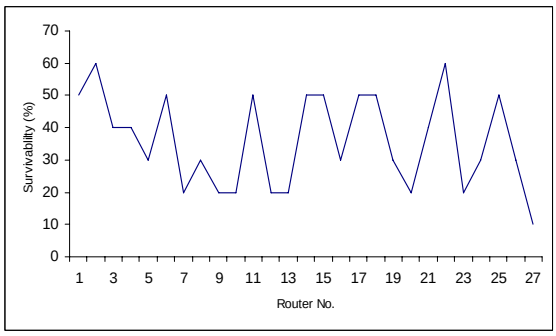


Fig. 12. Survivability metric for random attacker distribution

It can be observed from the results that all routers in the level are being affected by the attack traffic. However it is clear that since the attacking nodes are distributed in a random fashion, various routers are affected to a different extent. This is shown by varying levels of survivability and delay as shown in the results.

Based on the results of our simulations, it is believed that a network monitor mechanism may be useful for finer attack mitigation controls within a network that is under attack.

5. CONCLUSION

Two server-centric rate-limiting mechanisms were discussed; namely the pushback mechanism, as well as the level-k max-min mechanism. It was proposed that a network monitor could be used to enhance the performance of defences against DDoS attacks, through the notion of cooperative security management techniques. Routers that employ rate-limiting k-hops away from a server can easily be in different network arms; cooperative actions by the various network monitors will enable an exchange of information on the effectiveness of the defence mechanisms. While the primary purpose of rate-limiting is to protect the server, it is desirable that the unnecessary dropping of network packets, especially those coming from legitimate users, can be minimized.

A comparison resulted in the level-k max-min mechanism being chosen for our discussion. The simulation model was extended to include a network monitor that monitors congestion levels at routers where rate-limiting is in place. Using survivability and delay metrics that were defined, the simulation results have shown that it is possible to positively identify clear distinctions in router congestion levels during a DDoS attack whereby attacking nodes are unevenly distributed in the network topology.

6. ACKNOWLEDGEMENTS

This authors' work was supported by the National University of Singapore Academic Research Fund under grant R-252-000-118-112.

7. REFERENCES

[1] C. Douligeris, A. Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art", Computer Networks: The International Journal of Computer and Telecommunications Networking, 2004, 44(5): 643–666.

[2] R. Mahajan, S. M. Bellovin, et al, "Controlling High Bandwidth Aggregates in the Network", ACM SIGCOMM Computer Communication Review, 2002, 32(3): 62–73.

[3] Minuteman Software, GPSS World Reference Manual, Fourth Edition, Holly Springs, NC, U.S.A, 2001.

[4] K. K. Sung, J. M. Byoung, et al, "Cooperative Security Management Enhancing Survivability Against DDoS Attacks", In Proceedings of ICCSA 2005, (Singapore, May 9th – May 12th, 2005) Springer-Verlag, Berlin, 2005, 252–260.

[5] D. K. Y. Yau, J. C. S. Lui, et al, "Defending Against Distributed Denial-of-Service Attacks with Max-min Fair Server-centric Router Throttles", IEEE/ACM Transactions on Networking, 2005, 13(1): 29–42.

Managing Neighbor Replication Transactions in Distributed Systems

Noraziah Ahmad
Faculty of Comp. System & Software Engineering, University College of Engineering & Technology Malaysia,
Kuantan, Pahang 25000, Malaysia.
Email: noraziah@malaysia.com
Mustafa Mat Deris
Faculty of Information & Technology Multimedia, University College of Technology Tun Hussein Onn Group, Batu Pahat, Johor 86400, Malaysia.
Email: mmustafa@kuittho.edu.my
Norhayati Rosli
Faculty Chemical Engineering & Natural Resources, University College of Engineering & Technology Malaysia,
Kuantan, Pahang 25000, Malaysia.
Email: norhayati@kuktem.edu.my
Md. Yazid Mohd Saman, Rabiei Mamat, **W. N. Wan Nik Shuhadah**
Faculty of Science and Technology, University College of Science & Technology Malaysia,
Kuala Terengganu, Terengganu 21030, Malaysia.

ABSTRACT

This paper presents a new design to manage neighbor replication for distributed database system. We address how to build reliable system for managing transaction on Neighbor Replication Grid (NRG) in terms to preserve the data consistency and support advance fault-tolerant. We first recall the model and technique of NRG that impose neighbors binary vote assignment to its logical grid structure on data copies. Based on this model, we extent our work to design system for managing NRG transaction both in normal and failure cases. We present a result of correctness study on the system and finally an implementation of the system.

Keywords: Distributed Database, Replication, Consistency Transaction Model, Transaction Processing and Correctness.

1. INTRODUCTION

With the proliferation of computer networks, PCs and workstations, new models for workplaces are emerging [1]. In particular, organizations need to provide current data to users who may be geographically remote and to handle a volume of requests of data distributed around multiple sites in distributed database environment. Therefore, the storage, availability, accessibility and consistency are important issues to be addressed in order to allow distributed users efficiently and safely access data from many different sites. One way to provide access to such data is through replication. Thus, there is a need for minimal replication in order to provide an efficient way to manage the data and minimize the storage capacity.

Many vendors have adopted asynchronous solutions for managing replicated data such as Oracle 7 [2] and Lotus Notes. Lotus Notes works reasonably well for single object updates, but it fails when multiple objects are involved in a single update. While, Oracle7 uses a two-tier replication technique, but, inconsistencies may arise and a variety of reconciliation rules are required to merge the conflicting updates. Also, these products do not ensure serializability [3].

Another solution is based on synchronous replication [3, 4]. It is based on quorum to execute the operations with high degree of consistency [4] and also to ensure serializability. Synchronous replication can be categorized into several schemes, i.e., all data to all sites (full replication), all data to some sites and some data to all sites. However, full replication causes high update propagation and also needs high storage capacity [5,6,7]. Several studies model partial replication in a way that each data object is replicated to some of the sites. However, it is still undefined which copies are placed on which site, such that different degrees of quality of a replication scheme can be modeled. A few studies have been done on partial replication techniques based on some data items to all sites using tree structure technique [8, 9]. This technique will cause high update propagation overhead. Thus, some-data-items-to-all-sites scheme is not realistic. Furthermore, in many applications, there is update-intensive data, which should be replicated to very few sites. The European DataGrid Project [10] implemented this model to manage the file-based replica. It is based on the sites that have previously been registered for replication. This will cause the inconsistence number of replication occurs in the model. Also, the data availability has very high overhead as all registered replicas must be updated simultaneously. Thus, an optimum number of sites to replicate the data are required with non-tolerated availability of the system.

A new replication technique called Neighbor Replication on Grid (NRG) has been proposed in our previous work [11]. The NRG is treading a new path in replication that helps to maximize the system availability with low communication cost. We have intensively discussed this technique which consider only neighbor obtain a data copy (neighbor are assigned with binary vote one, and zero otherwise). This is due to the minimum number of quorum size required. Moreover, it can be viewed as an allocation of replicated copies such that a copy is allocated to a site if and only if the vote assigned to the site is one. This approach is particularly useful for large systems and also for P2P environment. However, the paper has not discussed the management of transactions.

Transaction Management is a well established concept in

* The work was supported by the IRPA Grant No. 54054.

database system research. A transaction can be defined as a sequence of operations over an object system and all operations must be performed in such a way that either all of them execute or none of them do. Transactions are used to provide reliable computing systems and a mechanism that simplifies the understanding and reasoning about programs. A halt in the operation of computer or other system can cause the transaction failure during its execution. In the event of disaster, some changes may have been made and others are not. This will jeopardize the data consistency and may produce incorrect results. Therefore, managing transaction is very important in order to preserve the data consistency and the reliability of the systems.

In this paper, without loss of generality, the terms node and site will be used interchangeably. An alternative and potentially more simplified approach to data distribution is provided by a replication server, which handles the replication of data to remote sites. In an NRG implementation, files have been used to represent data items since they are prone to have less locking information. Besides, data replication such as file replication is a key technology to keep important data available in the face of node failures [12].

The rest of the paper is organized as follows. In Section 2, we recall NRG model and technique. We introduce the transaction model, failure semantics and transaction processing in Section 3. Section 4 shows the implementation of NRG. Section 5 presents a result of a correctness study of the propose transaction model. Section 6 concludes the paper.

2. SYSTEM MODEL

In this section, we review the Neighbor Replication on Grid (NRG) model. All sites are logically organized in the form of a two-dimensional grid structure. For example, if an NRG consists of nine sites, it will logically organize in the form of 3 x 3 grid as shown in Fig. 1. Each site has a *master* data file. In the remainder of this paper, we assume that replica copies are data files. A site is either operational or failed and the state (operational or failed) of each site is statistically independent to the others. When a site is operational, the copy at the site is available; otherwise it is unavailable.

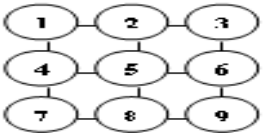


Fig. 1. A grid organization of 9 copies of an object.

Definition 1: A site X is a neighbor to site Y, if X is logically located adjacent to Y.

A data will be replicated to the neighboring sites from its primary site. The number of data replication, $d \leq 5$. For example, from Fig. 1. , data from site 1 will replicate to site 2 and site 4 which are its neighbors. Site 5 has four neighbors, which are sites 2, 4, 6, and 8. As such, site 5 has five replicas. For simplicity, the primary site of any data file and its neighbors are assigned with vote one and vote zero otherwise. This vote assignment is called binary vote assignment on grid. A neighbor binary vote grid assignment on grid, B, is a function such that $B(i) \in \{0, 1\}$, $1 \leq i \leq n$ where $B(i)$ is the vote assign to site i. This assignment is treated as an allocation of replicated copies and a vote assigned to the site results in a copy allocated at the neighbor. That is, 1 votes $\equiv$ 1 copy.

Let $L_B = \sum_{i=1}^{n} B(i)$ where, L_B is the total number of votes assigned to the primary site and its neighbors and it also equals to the number of copies of a file allocated in the system. Thus, $L_B = d$.

Let r and w denote the read quorum and write quorum, respectively. To ensure that the read operation always gets up-to-date value, $r + w$ must be greater than the total number of copies (votes) assigned to all sites. The following conditions are used to ensure the consistency:

1) $1 \leq r \leq L_B$, $1 \leq w \leq L_B$,
2) $r + w = L_B + 1$.

Conditions (1) and (2) ensure that there is a nonempty intersection of copies between every pair of read and write operations. Thus, the conditions ensure that a read operation can access to most recently updated copy of the replicated data. Timestamps can be used to determine which copies are most recently updated.

Let $S(B)$ is the set of sites at which replicated copies are stored corresponding to the assignment B. Then,

$S(B) = \{i \mid B(i) = 1, 1 \leq i \leq n\}$.

Definition 2: For a quorum group is any subset of $S(B)$ whose size is greater than or equal to q. The collection of quorum group is defined as the quorum set.

Let $Q(B,q)$ be the quorum set with respect to the assignment B and quorum q, then $Q(B,q) = \{G \mid G \subseteq S(B) \text{ and } |G| \geq q\}$. For example, from Fig. 1, let site 5 be the primary site of the primary data file x. Its neighbors are sites 2, 4, 6 and 8. Consider an assignment B for the data file x, such that

$B_x(5) = B_x(2) = B_x(4) = B_x(6) = B_x(8) = 1$

and $L_{B_x} = B_x(5) + B_x(2) + B_x(4) + B_x(6) + B_x(8) = 5$.

Therefore, $S(B_x) = \{5,2,4,6,8\}$.

If read quorum for data file x, r =2 and write quorum $w = L_{Bx}\text{-}r+1=4$, then the quorum sets for read and write operations are $Q(B_x, 2)$ and $Q(B_x, 4)$, respectively, where

$Q(B_x,2) = \{\{5,2\},\{5,4\},\{5,6\},\{5,8\},\{5,2,4\},\{5,2,6\},\{5,2,8\},$
$\{5,4,6\},\{5,4,8\},\{5,6,8\},\{2,4,6\},\{2,4,8\},\{2,6,8\},$
$\{4,6,8\},\{5,2,4,6\},\{5,2,4,8\},\{5,2,6,8\},\{5,4,6,8\},$
$\{2, 4, 6, 8\}, \{5, 2, 4, 6, 8\}\}$ and

$Q(B_x,4) = \{\{5,2,4,6\},\{5,2,4,8\},\{5,2,6,8\},\{5,4,6,8\},\{2,4,6,8\},$
$\{5, 2, 4, 6, 8\}\}$.

3. TRANSACTION MANAGEMENT

NRG Transaction Model

Transactions may return successfully or fail when any invoking set of transaction request to update the data object. There may be several reasons for the failure, such as "object not free", "server error" or "communication error". On receiving an "object not free" or a "server error", we are sure that transaction was not performed. However, on receiving a "communication error", we do not know the error occurred before or after the propagating transaction arrived at the destination host, so we do not know either the transaction was performed or not.

First we define the following:

a) T is a transaction.

b) D is the union of all data objects managed by all

transactions *T* of NRG and *x* represents one data object (or data file) in *D* to be modified by an element of T_α and T_β.

c) *Target set = {-1, 0, 1}* is the result of transaction *T* (see Table 1).

d) $\gamma = \alpha$ or β where it represents different group for the transaction *T* (before and until get quorum). μ represents the feedback from neighbor site.

e) *NRG transaction elements* $T_\alpha = \{T_{\alpha x,q_r} \mid r = 1, 2, ..., k\}$ where $T_{\alpha x,q_r}$ is a *queued* element of T_α transaction.

f) *NRG transaction elements* $T_\beta = \{T_{\beta x,q_r} \mid r = 1, 2, ..., k\}$ where $T_{\beta x,q_r}$ is a *queued* element of T_β transaction.

g) Successful transaction at primary site $T(\gamma_{x,\,q1}) = 0$, where $\gamma_{x,q1} \in D$ (i.e., the transaction locked a data *x* at primary).

h) Transaction at the neighbor site $T_{\mu x,q1}$ exists if either $T_{\gamma x,q1}$ or $T'_{\gamma x,q1}$ exists, where $T'_{\gamma x,q1}$ is a transaction that is transformed from $T_{\gamma x,q1}$.

We define *primary replica* for a particular data object *x* as a replica that accepts clients request. Each replica of $S(B)$ can be a primary or a neighbor replica concurrently. Any replica $i \in S(B)$ can be chosen as the primary replica, while other replicas $j \in S(B)$ where $i \neq j$ are neighbors. When a transaction T_α requests a *master* data file *x* from any replica of $S(B)$, that replica will be the primary, while others will be the neighbor replica for processing T_α respectively. If other sets of transactions invoke to update *x* after T_α, we call these sets of transactions as T_β. When T_β obtain a lock from *master* data file *x* from any site of $S(B)$, which is different site of primary replica for processing T_α, that site became as the primary processing for T_β. Simultaneously, the primary processing for T_β also functions as neighbor replica for processing T_α and vise versa. Other sites of $S(B)$ that is neither primary replica for processing T_α nor primary replicas for processing T_β will function as neighbor replicas for processing $T_{\gamma x,q1}$, where $\gamma = \alpha, \beta$.

Let $S(B)$ be the set of replicas that replicated copies are stored corresponding to the assignment *B* for particular data object *x*, $S(B_x)$ = {m(*i*-1,*j*), m(*i*,*j*-1), m(*i*,*j*+1), m(*i*+1,*j*)}. Two sets of transactions, T_α request data object *x* from m (*i*,*j*) replica, while T_β request data object *x* from m(*i*-1,*j*) respectively. The m(*i*,*j*) replica functions as the primary replica for processing T_α, whereas m(*i*-1,*j*), m(*i*,*j*-1), m(*i*,*j*+1) and m(*i*+1,*j*) are neighbor replicas for processing $T_{\gamma x,q1} \in T_\beta$. Simultaneously, m(*i*-1,*j*) replica functions as the primary replica for processing T_β, while m(*i*,*j*-1), m(*i*,*j*+1), m(*i*+1,*j*) and m(*i*,*j*) are neighbor replicas for processing $T_{\gamma x,q1} \in T_\beta$. Both m (*i*,*j*) and m(*i*-1,*j*) replicas execute two different processing task concurrently. The m(*i*,*j*) replica is the primary replica for processing T_α and neighbor replica for processing T_β, whereas the m(*i*-1,*j*) replica is the primary replica for processing T_β and neighbor replica for processing T_α.

The NRG transaction model considers different sets of transactions T_α and T_β. T_α is a set of transactions that comes before T_β, while T_β is a set of transactions that comes after T_α. Queued element q_1 denotes the transaction of the first queue. We define the effects of NRG transaction as the processing of one data item of the transaction. Hence, we can abstract NRG as a mapping of the following *type*

$T : D \rightarrow$ Target Set

Table 1. Meaning of Target Set.

Target set	**Meaning**
0	This means that no failure occurred during NRG transaction's execution. By $T(\gamma_{x,\,q1}) = 0$, where $\gamma_{x,q1} \in D$ is the data object processed by the transaction at primary site and $\gamma = \alpha, \beta$. The transaction was successful (i.e., the transaction locked the data file *x* at primary). *Write counter* track this value. For neighbor site, $T(\mu_{x,\,q1}) = 0$, where $\mu_{x,q1} \in D$.
1	This means *accessing failure*. By $T(\gamma_{x,\,t1}) = 1$, we mean that the destination server could not perform the job. Data file *x* managed by the primary site is already locked. The transaction has not executed. For neighbor site, $T(\mu_{x,\,q1}) = 1$, where $\mu_{x,q1} \in D$.
-1	This means *unknown status*. By $T(\mu_{x,\,q1}) = -1$, we mean that the neighbor site cannot tell if the NRG transaction has or has not been executed yet. This could happen when the destination host is down, or the link between primary and neighbor site is down, or both of the situations. In that case, the NRG request transaction or the message may be lost. So we do not know if the transaction has been executed or not at neighbor site. This will be tracked by *unknown status counter*.

All elements T_α and T_β may request data object *x* simultaneously at any site of $S(B)$ either at the same or different sites. In Figures 2(a), 2(b), 2(c) and 2(d), the details of the semantics of NRG transaction are illustrated. The semantic of NRG transaction *T* is that after issuing a set of transaction T_α, or many sets of transactions that include

T_α and T_β. All $T_{\gamma_{x,q_r}}$ try to get a lock of data *x*.

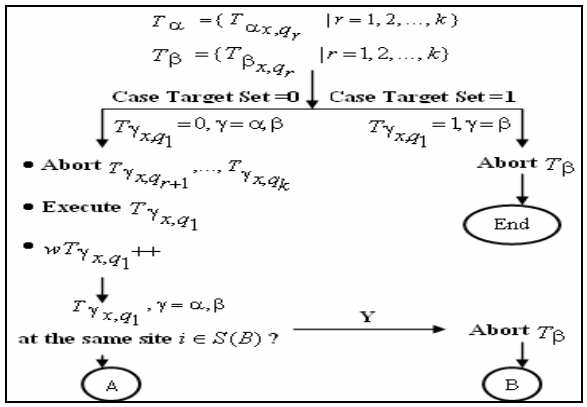


Fig. 2. (a) Initiate lock.

Case 1: Only one set of transaction T_α at any site $i \in S(B)$. For sites $S(B) = \{i \mid B(i) = 1, 1 \le i \le n\}$, $T_{\gamma_{x,q_1}} \in T_\alpha$ will get a lock, so that $T_{\gamma_{x,q_1}}$ will be executed while $T_{\gamma_{x,q_{r+1}}},\ldots,T_{\gamma_{x,q_k}}$ are aborted. $wT_{\gamma_{x,q_1}}$ is the write counter for $T_{\gamma_{x,q_1}}$ that increases when $T_{\gamma_{x,q_1}}$ gets a lock.

Case 2: Different sets of transactions where T_α and T_β request particular data object *x* at different site $i \in S(B)$. T_α and T_β can either come concurrently or not. Both $T_{\gamma_{x,q_1}}$ of $\gamma = \alpha, \beta$ get lock respectively. Let say all elements $T_{\alpha_{x,q_r}}$ and $T_{\beta_{x,q_r}}$ come to modify data object *x* at site1 and 2 respectively. $T_{\alpha_{x,q_1}}$ gets lock data object *x* at site 1 whereas $T_{\beta_{x,q_1}}$ gets lock data object *x* at site 2. $T_{\beta_{x,q_1}}$ is also the first queued element of T_α transaction ($T_{\alpha_{x,q_1}}$) at site 2.

Case 3: T_α and T_β request to update particular data object *x* at the same site $i \in S(B)$. Different sets of transactions $T_{\gamma_{x,q_1}} \in T\alpha$ and $T_{\gamma_{x,q_1}} \in T\beta$ may request the same data object *x* at site 1 and $T_{\alpha_{x,q_1}} \neq T_{\beta_{x,q_1}}$ at site 1. So, $T_{\gamma_{x,q_1}} \in T\beta$ has an accessing failure and is aborted while $T_{\alpha_{x,q_1}}$ will be executed. Note: We choose to abort $T_{\gamma_{x,q_{r+1}}},\ldots,T_{\gamma_{x,q_k}}$ and $T\beta$ because of the application constraint of the software *text editor* that result apparently successfully completed update operation by one user overridden by another user. This is not the restriction of our model but we force to abort those transactions in terms to preserve the data consistency.

When $T_{\gamma_{x,q_1}}$ gets a lock for a particular data object *x*, it will be propagated synchronously to neighbors site $i \in S(B)$ until it obtains the quorum. On detecting a *system failure* while propagating a lock, primary set $T_{u_{x,q_1}}$ be equal to -1.

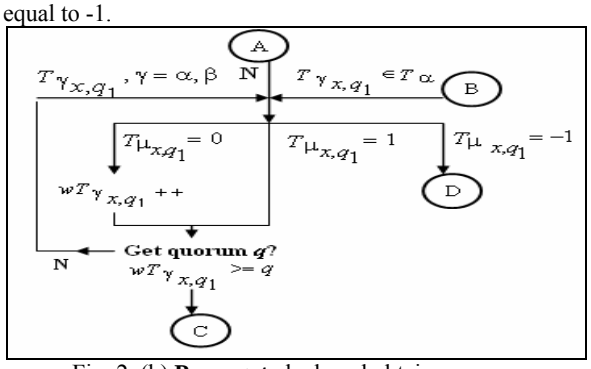


Fig. 2. (b) **Propagate** lock and obtain quorum.

The actual effect of $T_{\gamma_{x,q_1}}$ where $\gamma = \alpha, \beta$, depends on the majority quorum, which its primary replica processing can obtain from $T_{u_{x,q_1}}$ of neighbor replica. The primary of the first transaction that gets a quorum denoted as $T'_{\gamma_{x,q_1}}$ will survive while others are aborted.

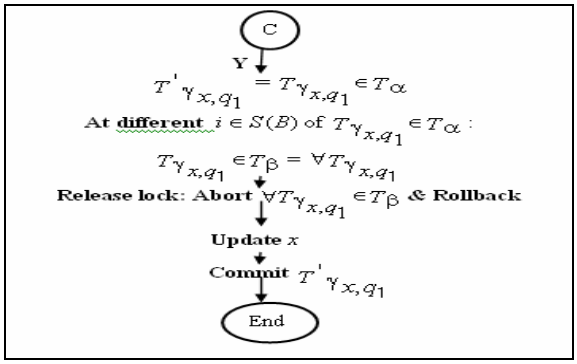


Fig. 2. (c) Release lock, update and commit.

There are two types of replica(s) failures include *primary* and *neighbor replica failure*. For the case of *primary replica failure*, NRG reconcile and do the conflict resolution. Otherwise, replica checks the unknown status counter and decides either to proceed $T'_{\gamma_{x,q_1}}$ or impose the timestamping method (more than a majority quorum has system failure).

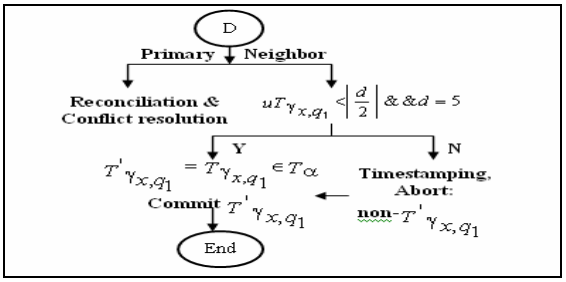


Fig. 2. (d) Failure (Unknown status).

$T'_{\gamma_{x,q_1}}$ will change the permission mode of data object *x* and then acknowledges the client for an updating process and commit the transactions changes.

Failures: There are three classes of failures in a replicated-server environment executing on NRG

transactions: *transaction abort, sites failure* and *links failure*. We impose *system failures* to denote the latter two failures and assume a single failure for system failures.

NRG Transaction Processing

In the NRG transaction processing, clients are simple sources of transactions. Three system components are involved in processing a transaction submitted by clients:

a) *NRG transaction manager* (NTM) accepts a set of transactions $T_{\gamma_{x,q_l}}$ where $\gamma = \alpha, \beta$ from the clients. Besides, NTM also accepts $T_{u_{x,q_l}}$ submitted by other replica $i \in S(B)$. NTM determines replica site function either to be primary or neighbor. Then, NTM calls relevant function to do the task of transaction processing.

b) A primary replica accepts transactions that are passed from its NTM and acts as the coordinator to manage the atomic $T_\alpha = \{T_{\alpha_{x,q_r}} \mid r = 1, 2, ..., k\}$ and $T_\beta = \{T_{\beta_{x,q_r}} \mid r = 1, 2, ..., k\}$. Primary replica also receives $T_{u_{x,q_l}}$ as the feedback from its neighbors.

c) Each neighbor replica $i \in S(B)$ of the data x accepts $T_{\gamma_{x,q_l}}$ or $T'_{\gamma_{x,q_l}}$ from the primary replica site $j \in S(B)$ where $i \neq j$.

4. IMPLEMENTATION

In this section, we present the implementation of the system. The purpose of this implementation is to illustrate that our system can be use in practical applications. The implementation requires both hardware and software components. Table 2. shows the server main components specification for each node in the cluster servers implemented.

Table 2. Server main components specification.

Hardware	Specifications
Processor	1.8 GigaHertz Intel Pentium IV
Memory	256 Megabyte DIMM
Chip Set	82845 GL HOSTS/AGP/Controller
Network Card	Built-In RTL810L chipset

The experiment was done in shell programming and Perl integrated with FTP services for agent communication. Linux OS: Red Hat Linux Kernel release 9 and Linux Slackware 2.4.2 are used as a platform to the replicated servers. The applications for users include *mcedit, vi* and *vim editor*.

An Application Example

To give a better intuition of how we manage the transaction through the NRG protocol, we develop the NRG daemon. From the users' perspective, the functionality offered by our transaction processing model is highly attractive where it suitable for large-scale system. Nonetheless, for an implementation we test NRG daemon in grid cluster server under local area network (LAN) with 9 nodes. Each server or node is connected to one another through a fast Ethernet switch hub. Fig. 3 shows a cluster with 9 replication servers that are logically connected to each other in the form of two-dimensional 3 x 3 grid structure. In the remainder of this paper, we assume that replica copies data files.

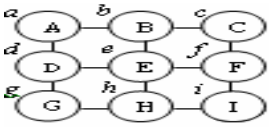


Fig. 3. A cluster with 9 replication servers.

By imposing neighbor binary voting assignment, each primary replica copies data *x* to its neighbor replicas. Client can access data *x* at any replication server *x* at $B_x(i) = 1$. Assume that primary data files *a* located on server A, primary data file *b* located on server B, and so on. Based on our transactional model, *x* for $T_{\gamma_{x,q_l}}$ represent any data file *a, b, c, d, e, f, g, h* and *i*.

Without lost of generality, we look at this simple case. Two different sets of transactions, $T_\alpha = \{T_{\alpha_{x,q_r}} \mid r = 1, 2, ..., k\}$ & $T_\beta = \{T_{\beta_{x,q_r}} \mid r = 1, 2, ..., k\}$ come to update data file *a* at replica A and B in the absence of system failures. The original implementation of NRG transactional model as follow: clients request to update data file *a* from primary replica A and B (refer case 2 in Section 3.1, where site 1 and 2 are the primary replica A and B respectively). Neighbor binary voting assignment is initiates as $B_a(A) = B_a(B) = B_a(D) = 1$ so that $S(B_a) = \{A, B, D\}$. Primary replica A with IP 192.168.100.21, primary replica B with IP 192.168.100.36 and primary replica D with IP 192.168.100.39 locate data *a, b* and *d*. Table 3. shows the Primary-Neighbor Grid Coordination (PNGC) for replica A, B and D.

Table. 3. PNGC for replica A, B and D.

A: 192.168.100.21	B: 192.168.100.36	D: 192.168.100.39
B: 192.168.100.36	D: 192.168.100.39	A: 192.168.100.21
D: 192.168.100.39	A: 192.168.100.21	B: 192.168.100.36

The NRG daemon for primary replica A, monitors all users current status that access particular data *a, b* and *d*. At the same time, the NRG daemon for primary replica B monitors all users' current status that access particular data *a, b, c* and *e*. If any user accesses the particular data *x*, then it redirects the user's information such as the *pid, user name, tty, log time* and *access editor* to the log information. The daemon manipulates the log information by using *awk utility*.

The NTM of primary replica A and B pass to their ***primary_replica_processing*** function for recognizing which transaction gets the lock. $T_{\gamma_{x,q_l}}$ is recognized based on its login time. Since several transactions come concurrently to access data *a*, the first queued element of data *a* in the log information will get the lock. At primary replica A, $T_{\alpha_{a,q_l}}$ gets the lock and server status is initiated to 1 for its Target Set. The NRG daemon kills *pid* of $T_{\alpha_{a,q_2}}, ..., T_{\alpha_{a,q_k}}$. Then, it broadcasts messages to acknowledge clients. The same procedure also occurs at primary replica B where the first queued element of T_β is recognizes through its log information. $T_{\beta_{a,q_l}}$ obtains a lock at primary replica B and server status is initiated to 1 for its Target Set. The NRG daemon broadcasts messages to clients after killing *pid*

of $T\beta_{a,q2},...,T\beta_{a,qk}$. Since replica A and B get the write lock, NRG daemon controls the mode permission. Therefore, other transactions cannot read or update data file *a* at that time.

Primary replica A and B propagate lock synchronously to its neighbor replicas based on Table 3. Primary replica processing for $T\alpha_{a,q1}$ propagates the locks to its neighbor replicas B and D. At the same time, primary replica processing for $T\beta_{a,q1}$ propagates the locks to its neighbor replicas D and A. Each NTM of neighbor replica calls ***neighbor_replica_processing*** function to check its feasibility lock and sends feedback to the primary.

The first transaction that obtains a quorum is denoted as $T'\gamma_{a,q1}$ and it releases other $T\gamma_{a,q1}$. In this experiment, $T\beta_{a,q1}$ obtains the quorum and become $T'\gamma_{a,q1}$. $T'\gamma_{a,q1}$ at primary replica B releases the lock of $T\alpha_{a,q1}$ at primary replica A. Therefore, $T\beta_{a,q1}$ gets all locks from $S(B_a)$. Next, NRG daemon changes the access permission mode of data file *a* at primary replica B and acknowledges client to update it. After client has finished updating the data file *a*, all replicas of S (B_a) commit $T'\gamma_{a,q1} \in T\beta$ ($T\beta_{a,q1}$) synchronously. Finally, the NRG daemon changes the access permission mode to unlock data file *a*. Table 4 shows the experiment result of how NRG handle concurrent transactions $T\alpha_{a,q1}$ and $T\beta_{a,q1}$ at primary replica A and B respectively.

Table 4. The experiment result of how NRG handle Concurrent transactions.

REPLICA	A	B	D
TIME			
t1	unlock(*a*)	unlock(*a*)	unlock (*a*)
t2	begin_transaction	begin_transaction	
t3	write lock(*a*) counter_w(*a*)=1	write lock(*a*) counter_w(*a*)=1	
t4	wait	wait	
t5	propagate lock:B	propagate lock:D	
t6	propagate lock:D		lock(*a*) from B
t7	propagate lock:B	get lock:D counter_w(*a*)=2	
t8	propagate lock:D	obtain quorum release lock: A	
t9	abort $T\alpha_{a,q1}$ & rollback, lock(*a*) from B		
t10		update *a*	
t11		commit $T\beta_{a,q1}$	
t12	unlock(*a*)	unlock(*a*)	unlock (*a*)

5. CORRECTNESS

In this section, we outline the informal analysis of the correctness.

Assertion 1: If the transaction gets all locks from replica $i \in S(B)$, then the transactions have been executed successfully.

Proof: The only way that a transaction gets a lock when $T\gamma_{x,q1} = 1$ with $T\gamma_{x,q1} \in T\alpha$. Then we need to abort $T\gamma_{x,qr+1},...,T\gamma_{x,qk}$ which are the elements in $T\alpha$. To get all locks, we must have $wT\gamma_{x,q1} \geq \lfloor d/2 \rfloor$. This means that the primary needs to get the majority locks from its neighbors. Each neighbor $i \in S(B)$ sends a feedback to the primary. If the primary gets a majority lock, it means that $T'\gamma_{x,q1} = T\gamma_{x,q1} \in T\alpha$ where $T'\gamma_{x,q1}$ gets a quorum. Next, the primary sends command "*Abort*" to other neighbor $i \in S(B)$ in the quorum but $T\gamma_{x,q1} \in T\alpha$ fails to obtain all locks. Consequently, when $T\gamma_{x,q1} \in T\beta$ releases its lock in, then $T\gamma_{x,q1} \in T\alpha$ gets the lock of neighbor $i \in S(B)$ for $\forall$ neighbor $i \in S(B)$. When a user finishes updating the data, $T'\gamma_{x,q1}$ commit to $\forall$ neighbor $i \in S(B)$. Therefore, all replica of $S(B)$ will perform and execute the transaction successfully.

6. CONCLUSION

Managing transactions on NRG provides advanced fault tolerance capabilities that allow it to withstand failure both in handling quorum locking and the transaction execution. We have implemented NRG in the form of logical design grid structure under LAN for mere simplicity to get a clearer algorithm. System failure such as server crash and link crash are considered in our implementation. Network partitions occur regularly due to either hardware (e.g. temporarily disconnected switches) or software (heavily loaded servers).

Nowadays, many systems cannot be fully utilized to handle high frequent update operation with less computational time for synchronous replication in large-scale system. Timeliness in synchronization has become show stopper to maximize the usage of system but at the same time contribute to the consistent and reliable computing. NRG resolve this challenge by alleviates lock with small quorum size before capturing update and commit transaction synchronously to the database that require the same update data. Our transaction is proportional to the quorum size. Furthermore, NRG guarantees the consistency since the transaction execution is equivalent to one-copy-serializability.

In summary, this paper has investigated a system for building reliable computing which combines the replication and transaction techniques. We extensively describe the NRG model, transactions model, and transaction processing model. Besides, we also present an application example for NRG implementation. Finally, the correctness analysis is then carried out for the propose transaction model.

In wide area networks (WAN), partitions can be common. A distributed replication system within a WAN is composed

of several subnets located about several or thousand miles away and connected by gateways. Data replication and distribution in WANs are fundamental problem for many important and popular infrastructures such as Internet edge services, grid computing and peer-to peer (P2P) network. As grid systems increase in scale, they began to require solution to issues on self-configuration, fault tolerance, scalability, for which peer-to-peer research has much to offer. Peer-to-peer systems, on the other hand, focus on dealing with instability transient population, fault tolerance and self-adaptation [14]. In summary, one can say that, "Grid computing addresses infrastructure, but not yet failure, while peer-to-peer address failure, but not yet
Infrastructure" [14, 15]. We will consider NRG in WAN for the future work.

REFERENCES

[1] O. Wolfson, S. Jajodia, et al, "An Adaptive Data Replication Algorithm", ACM Transactions on Database Systems, 1997, 22(2):255-314.

[2] Oracle, Oracle7 server Distributed Systems: Replicated Data, Redwood Shores. Available at http://lina.cli.di.unipi.it:8000/WG73-doc/server/sd273/toc/html, 2002.

[3] J. Holliday, R. Steinke, et al "Apidemic Algorithms for Replicated Databases", IEEE Trans. On Know. and Data Engineering, 2003,15(3):1-21.

[4] H. Stockinger, "Distributed Database Management Systems and The Data Grid", IEEE-NASA Symposium, 2001, 1-12.

[5] Budiarto, S. Noshio, et al , "Data Management Issues in Mobile and Peer-to-Peer Environment", Data and Knowledge Engineering, Elsevier, 2002, 41:183-204.

[6] L.Gao, M. Dahlin, et al, "Improving Availability and Performance with Application-Specific Data Replication", IEEE Trans. On Knowledge and Data Engineering, 2005,17(1):106-120.

[7] M.Mat Deris, J.H. Abawajy, et al, "An Efficient Replicated Data Access Approach for Large Scale Distributed Systems", IEEE/ACM Conf. On Cluster Computing and Grid (CCGRID2004), Chicago, USA,2004.

[8] M. Nicola, M. Jarke, "Performance Modeling of Distributed and Replicated Databases", IEEE Trans. On Knowledge and Data Engineering, 2000, 12(4):645-671.

[9] J.Huang, Q.F. Fan, et ale, "Improved Grid Information Service Using The Idea of File-Parted Replication", Lecture Notes in Computer Science Springer, 2005,3584:704-711.

[10] P. Kunszt, E. Laure, et al, "File-based Replica Management", International Journal of Future Generation of Computer Systems, Elsevier, 2005,21:115-123.

[11] M. M. Deris, D. J. Evans, et al, "Binary Vote Assignment on Grid For Efficient Access of Replicated Data", Int'l Journal of Computer Mathematics, Taylor and Francis, 2003,80(12):1489-1498.

[12] I.R.Chen, D.C.Wang, et al "Analyzing User-Perceived Dependability & Performance Characteristics of Voting Algorithms for Managing Replicated Data", Distributed and Parallel Databases, Kluwer Academic Publisher, 2003, 199-219.

[13] B. Bhargava, "Concurrency Control in Database Systems", IEEE Trans. Knowledge and Data Engineering, 1999,11(1):3-16.

[14] S.A. Theotokis,D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies", ACM Computing Surveys, 2004,36(4):335-371.

[15] Foster , A. Iamnitchi, "On Death Taxes and the Convergences of Peer-to-Peer and Grid Computing", Proceeding of the Int'l Workshop on Peer-to-Peer System (IPTP'03), 2003, 335-368.

A Simple Distributed Broadcasting Algorithm for Ad Hoc Networks*

Sadia Aziz, Li La yuan, Sun Qiang
School of Computer Science, Wuhan University of Technology, Wuhan, Hubei 430063, China
s_aadia2002@hotmail.com, jwtu@public.wh.hb.cn, chyang_sun@163.com

ABSTRACT

In this paper, a simple distributed heuristic-based algorithm is presented for Ad Hoc Networks. The algorithm is based on joint distance-counter threshold scheme but adaptive in nature. It runs in a distributed manner by each node in the network without needing any global information. Each node in an ad hoc network hears the message from its neighbors and decides whether to retransmit or not according to the signal strength and the number of the receiving messages. By using the modified covering problem, it's easy to find the nodes that consist of the vertices of the Hexagonal lattice to cover the whole networks.

Keywords: Ad Hoc Networks, Hexagonal lattice

1. INTRODUCTION

A mobile ad hoc network is a self-organizing network, which is used in disaster rescues, wireless conferences in the hall, battlefields, monitoring objects in a possibly remote or dangerous environment, wireless Internet etc.

The research of routing is still at the beginning and some routing protocols have been put forward [1-8] in mobile ad hoc networks. Most of these protocols depend on a broadcast mechanism [9-11]. Various efficient broadcast mechanisms have been proposed for MANET but they mostly depend upon the topology or neighborhood information. There are also many distributed broadcasting algorithms with no use of global information have been proposed for use. But so far, no generic framework can capture a large body of distributed Broadcast Algorithm. An efficient adaptive broadcast algorithm based on joint dynamic distance-counter threshold is proposed in this paper. It drastically reduces the effect of the mobility and no exchanged messages or control messages are needed. Joint distance-counter threshold is to provide both a satisfied coverage, less broadcast and average latency, unlike joint distance threshold just guaranteeing a high coverage, and counter threshold just guaranteeing a high saved broadcast and less latency.

The rest of this paper is organized as follows. Section 2 gives an overview of key efficient flooding protocols proposed for ad hoc networks. In Section 3 definitions and assumptions are given. Section 4 presents Adaptive joint distance-counter threshold broadcast algorithm. Simulation results are in Section 5 and conclusions are drawn in Section 6.

* This work is proudly supported in part by the Grand Research Problem of the National Science Foundation of P.R.C. under Grant No. 90304018

2. RELATED WORK

The broadcast problem has been extensively studied for Ad hoc networks. The Multipoint Relay (MPR) method [12] [14] is presented for efficient flooding in mobile wireless networks. In [12], a node periodically exchanges the list of adjacent nodes with its neighbors, each node collects *2-hop* neighbor information, and selects the minimal subset of forwarding neighbors that covers all neighbors within *2-hop* away. Only nodes chosen as forwarding neighbors rebroadcast the flooding packet. In [13], two practical heuristics for selecting the minimum number of forwarding neighbors are proposed. In order to choose forwarding neighbors, a node needs to know its *1-hop* and *2-hop* neighbors. Any changes from any neighbors can cause a reselection of forwarding neighbors. Thus, the exchanged messages are main overhead to the algorithm.

[15] proposed a broadcast protocol for ad hoc networks based on a distributed hierarchical framework i.e. cluster structure. With a cluster platform, the broadcast protocol in [15] can choose a dominating set, which is only composed of cluster head nodes and gateway nodes to rebroadcast packets. In [16], using a passive clustering, clustering algorithm is suspended until the data traffic commences. Thus it reduces setup latency and control overhead caused by active clustering. The main drawback of using cluster structure is its significant communication overhead for maintaining the structure in a moving environment [17]. Ivan Stojmenovic et al. [18] proposed a simple and efficient distributed algorithm for calculating connected dominating sets enhanced by neighbor elimination scheme and highest degree key. Nevertheless location information of each node should be available to implement such algorithm.

Note that the global information is difficult and sometimes is impossible to get in ad hoc networks; broadcast algorithms based on global network information cannot provide scalability and is unsuitable for the qualifications of mobile ad hoc networks. In [19], a generic distributed broadcast scheme is proposed, which depends on correct *k-hop* neighborhood information to reduce the broadcast storm problem and guarantee full coverage. If the neighborhood information cannot be acquired correctly, the algorithms in [12-21] cannot run properly. In order to acquire correctly neighborhood information, more exchanged messages are needed, which will deteriorate the performance of networks and

consume many of limited resources in ad hoc networks.

The distance-based algorithm and the counter-based algorithm in [11] give a simple scheme, in which each node, when receiving packets from its neighbors, decides whether to forward the broadcast packet or not according to distance or counter threshold. But all these methods are in dilemma about how to provide both high coverage, less rebroadcast and average latency.

3. FORMAL DEFINITIONS AND SYMBOLS

An ad hoc network can be mapped to a unit disk graph $G(t)=(V,E)$, where V is the set of nodes and E is the set of logical edges at which two nodes are connected if their geographical distance is within a given transmission range r. Considering the effect of the mobile nodes, $G(t)$ is a time-relevant function.

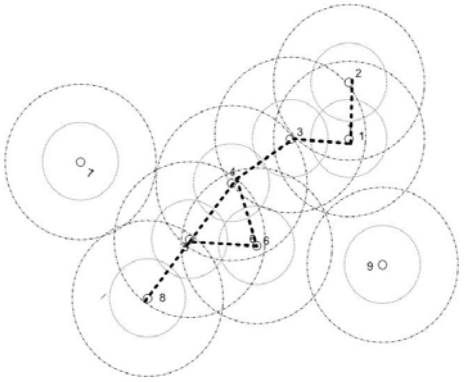


Fig. 1. An ad hoc network

The symbols and definitions used in this paper are defined as follows:

Definition 1: $d(u,v)$ is defined the distance between node u and v within their transmission range r , $d(u,v)\le r$.

Definition 2: $c(u)$ denotes the number of received messages in node u during broadcasting.

Definition 3: dn is the distance of a node from its neighbor.

Definition 4: d is the distance between a node and source node.

Definition 5: D_{Th} denotes a distance threshold, which is calculated by using d and constant α $D_{Th}=\alpha$ d where $0\le D_{Th}<r$.

Definition 6: Dpth denotes the previous calculated distance threshold

Definition 7: C_{Th} denotes a counter threshold, where $C_{Th}\ge 0$.

Definition 8: $N(u)$ is a set of neighbors of node u, $v\in N(u), d(u,v)\le r$. See Figure 1, the neighbors of node 1 consist of node 2, 3 represented as $N(1)=\{2,3\}$ or the neighbors of node 5 consist of node 4,6,8 represented as $N(5)=\{4,6,8\}$

Definition 9: $I(u)$ is a subset of $N(u)$, $v\in I(u), d(u,v)\le D_{DTh}\le D_{Th}$. In Fig.1, $I(1)=\Phi$.

Definition 10: $E(u)$ is a subset of $N(u)$, $v\in E(u),\left(D_{DTh}<d(u,v)\le r\right)\cup\left(D_{Th}<d(u,v)\le r\right)$.

See Figure 1, $E(1)=\{2,3\}$.

Definition 11: $Rt(u)$ is a set of nodes that retransmit the message from source node,

$$Rt(u)=\left\{u\mid u\in\left\{\begin{matrix}\{E(u_1)\cap E(u_2)\cap\cdots E(u_k)\}\cup\\ \{u\notin\{E(u_1)\cap E(u_2)\cap\cdots E(u_k)\},c(u)<C_{TH}\}\end{matrix}\right\},u_i\in V\right\}$$

.

The following assumptions are made in our system model. Mobile nodes in an ad hoc network share a single common channel, and a message transmitted by a node reaches all its neighbors at the same time. The maximum transmitting radius of each node in the network is the same. There are no unidirectional links. The broadcast messages do not require an explicit acknowledgement to confirm the reception.

4. ADAPTIVE JOINT DISTANCE-COUNTER THRESHOLD(AJDCT) BROADCASTING ALGORITHM

In this section, an efficient distributed heuristic-based algorithm is presented which covers the shortcomings of JDCT (Joint Distance-Counter Threshold). In JDCT if the transmission node is surrounded by many node and distance of all these nodes are less than threshold distance then every node would be in counter threshold mode and the other nodes of network will face the coverage problem. To alleviate this problem in this algorithm we calculate threshold distance dynamically. The proposed algorithm aimed at solving the broadcast storm problem without consuming additional network resources, such as bandwidth and energy.

Algorithm

Our approach to alleviate Broadcast storm problem is to intelligently inhibit some hosts from rebroadcast to reduce redundancy, contention, and collision and to get better coverage. In order to alleviate the broadcast storm problem, $Rt(u)$ has to be found by an efficient distributed heuristic-based algorithm. The relationship between the redundancy of a broadcast and the additional coverage is revealed in [11].

Moreover, it also shows the relationship among the additional coverage, $d(u,v)$ and $c(u)$ i.e. the farther $d(u,v)$ is, the larger additional coverage of a node can be acquired, and the larger $c(u)$ is, the smaller additional coverage of a node can be acquired. [12] tries to get the large additional coverage and avoid the worst saved broadcast and more latency time at the same time. But in some situations it can't get the required results. We have proposed an intelligent algorithm, which also works in these situations.

Each broadcast packet an additional field of distance in its header. Whenever a node transmits a broadcast packet, this field has the information of distance between the transmitting node and the source from which it received transmission.

The algorithm works as follows. When a node *u* sends a broadcast message, its neighbors will receive *message* and compute $d(u,v)$ according to the signal strength [24]. Let x is the neighbor node of v and hears message from v, It'll calculate the distance threshold from d(u,v)., $D_{th}=\alpha$ d, where $0.7<\alpha<0.8$.(The distance threshold would be calculated by every node would be compared from the previous calculated distance threshold. we'll take the distance threshold which would be larger). Now x will compute its distance from v and compare with the calculated distance threshold. If the d_n is less than D_{th} then

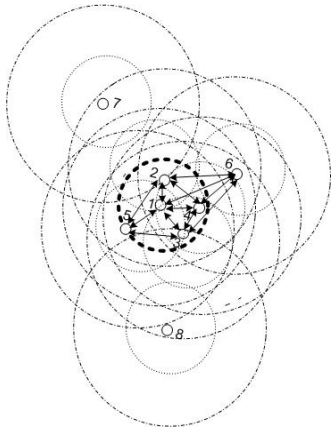
If $x \in \{E(u_1) \cap E(u_2) \cap \cdots E(u)\}$ or $x \notin \{E(u_1) \cap E(u_2) \cap \cdots E(u)\}, c(u) < C_{Th}$, $\{u_1, u_2, \cdots u\} \subseteq N(x)$,it easy to get that $x \in Rt(u)$ and node x waits for a short time. The delay helps to avoid nodes transmitting message all at once. If node v didn't receive any messages during this short delay, it will transmit message when time expires. Otherwise, it will compute the distance from the sending node and wait again. If $c(v) \geq C_{Th}$, node v stops transmitting. The algorithm can be described as follows:

1. Initialize $D_{pth}= 0$, $C_{Th} = C$ (C =1,2…6)
2. If a message "msg" is heard by a node other than the source node, then calculate the distance "d" between that node and the source node
3. Use the function given in section 4.1 to calculate the $D_{th.}$
4. Compare the D_{th} with its distance d_n from its neighbors to decide whether it is in the range D_{th} or not. If $d_n<D_{th}$. Proceed to C1. Else go to F1.

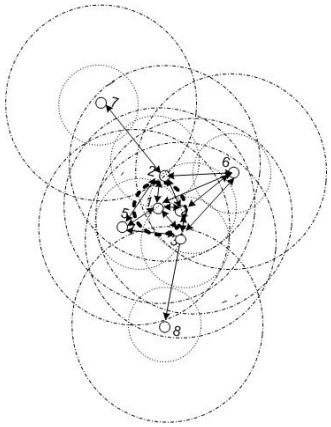
C1. Initialize the counter c=1 when the broadcast message is heard for the first time. In F1 if the message is heard again, interrupt waiting and go to C2, else go to F1.
C2. c=c+1. If $c<C_{th}$ resume the interrupted waiting in F1. Else go to F2.
F1. Wait for the short time determined by distance-counter function. If the message is heard again during waiting then if the node is in DTM (Distance Threshold Mode), go to 2, else go to C2. Else submit for transmission and wait until transmission actually starts.
The message is on air.
Exit
F2. Cancel the transmission of msg if it was submitted in F1.
The host is inhibited from rebroadcasting message.
Exit

In JDCT if the source node is surrounded by nodes such that the distance of each node from the source node is less than Distance threshold then all the surrounding neighbor nods will depend on the counter threshold for further retransmission and will face the shortcomings (coverage problem) of counter threshold based scheme. Our Algorithm overcomes this problem by calculating the distance threshold dynamically.

As in Figure.2 we can see that dynamic distance threshold provides a better reachable than JDCT because there is no fixed distance threshold. In some situations AJDCT has the same performance but in some situation it has better Performance than JDCT. We have proved this better performance by simulation results.



(a)



(b)

Fig. 2. Comparison for redundant to broadcast among (a) JDCT algorithm (b) AJDCT algorithm

The comparison with the JDCT and AJDCT will be given through an example in Figure 2.

First some assumptions are given as follows.

The sequence of retransmission is decided by its distance to the source node. The larger the distance is, the earlier the node may transmit. From the Fig we can see 6 is the source node and the sequence of retransmission from the source node is 6-1-2-3-4 because d(6,1)> d(6,2)> d(6,3)> d(6,4). By this example we'll show that JDCT Algorithm can't cover the nodes 7 and 8 but ADJCT can cover these nodes.

1. We assume that $C_{th}=3$
2. D_{th} is dynamic and it'll be calculated at the run time Now from the aforementioned definition

$N(1) = E(1) = \{2,3,4,5,6\}, I(1) = \Phi$.

$N(2) = E(2) = \{1,2,3,4,5,6,7\}, I(2) = \Phi$.

$N(3) = E(3) = \{1,2,3,4,5,6,8\}, I(3) = \Phi$.

$N(4) = E(4) = \{1,2,3,4,5,6\}, I(4) = \Phi$.

$N(5) = E(5) = \{1,2,3,4\}, I(5) = \Phi$

$N(6) = E(6) = \{1,2,3,4\}, I(6) = \Phi$

$N(7) = E(7) = \{2\}, I(7) = \Phi$

$N(8) = E(8) = \{3\}, I(8) = \Phi$

We don't know about the threshold of any node before the transmission starts. After the transmission start every node will calculate threshold in distribute manners. So we assume that in the start I= Φ and E=N. In Fig. 2.a, each node uses the JDCT algorithm to decide whether to retransmit or not. The result match the analysis in [25]

In Figure. 2.b, each node uses the AJDCT algorithm to decide whether to retransmit or not. The nodes decide as follows,

Step 1: Node 6 is the source node that starts transmission, which is received, by neighbor nodes 1,2,3,4.

Step 2: Node 1 is the furthest node among the neighbor nodes of the node 6, so node 1 retransmits M first.

Step 3: As $N(1) = E(1) = \{2,3,4,5,6\}, I(1) = \Phi$, so when node 1 retransmits first of all node 5 will hear node first and calculate D_{th} according to algorithm. So according to algorithm

$N(1) = \{2,3,4,5,6\}, E(1) = \{2,3,5\}, I(1) = \{4\}, c(4) = 1$ for d (1,5)> d (1,3)> d (1,2), node 5 will retransmit the msg in the next step.

Step 4: When node 5 retransmits the message, all its neighbors (node 2, 3, 4) will received the message and calculate D_{th} again. And we can get

$N(5) = \{1,2,3,4,6\}, E(5) = \{1,2,3,4,6\}, I(5) = \Phi, c(4) = 2$

and as d (5, 2)> d (5, 3), node 2 will retransmit the message in the next step.

Step 5: When node 2 retransmits the message, we can get $N(2) = \{1,3,4,5,6,7\}, E(2) = \{3,4,5,6,7\}, I(2) = \Phi, c(4) = 3 > C_{Th}$ as $c(4) = 3 > C_{Th}$, node 4 is saved and node 7 is covered by node 2.

Step 6: When time expires, node 3 will retransmit the message and node 8 will be covered.

5. SIMULATION RESULTS

Simulations are performed to evaluate the new broadcasting algorithm and compare with other existing algorithms. A Mobility Framework for OMNeT++ is used as a tool. The network possesses 100 nodes in a 1000×1000 meter square. Nodes in the simulation move according to "random waypoint" model. The transmitting radius of each node is 230 meters and channel capacity is 10Kbits/sec. The mobility speed of a node is set from 0 m/s to 30 m/s. The CSMA/CA is used as the MAC layer in our experiments. Four distributed broadcasting algorithms are compared, as following.

- **SB:** straightforward broadcasting algorithm
- **JDCT:** JDCT broadcasting algorithm
- **AJDCT: A**daptive JDCT broadcasting algorithm

The performance measures of interest are:

- ***Average latency***: defined as the interval between its arrival and the moment when either all nodes have received it or no node can rebroadcast it further.
- ***Ratio of Saved Rebroadcast (RSR)***: the total number of saved rebroadcast nodes is divided by the total number of nodes receiving the broadcast message.
- ***Coverage***: the number of nodes that receive broadcast packets divided by the total number of nodes.

In order to compare the performance of the algorithm, we use parameters in our algorithm as same as in [25]. That is, $D_{Th} = 0.9r, C_{Th} = 3$. The first set of experimental results Fig .3. Demonstrates the coverage of different algorithms. Notice that SB cannot guarantee 100% coverage because of the existing hidden/exposed terminals. As we use a dynamic threshold scheme, AJDCT can get a higher coverage than JDCTdoes.

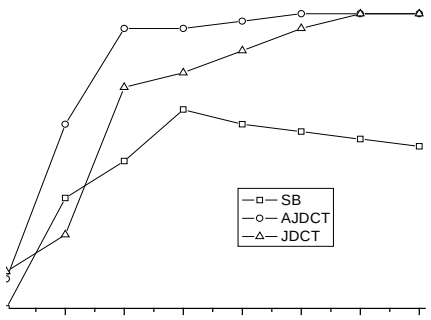


Fig. 3. Coverage of different Algorithms

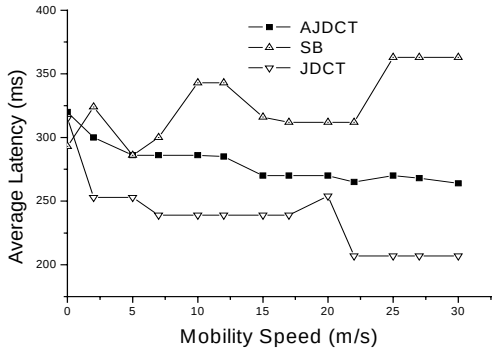


Fig. 4. Average Latency vs. Mobility Speed

The average latency using different broadcast algorithms with varying node speeds is reported in Fig. 4. Fig. 5 shows the ratio of saved rebroadcast using different algorithms with varying node speeds (from 0 to 30m/sec). In AJDCT, because more nodes are selected to retransmitting than those in JDCT, the average latency and the ratio of saved rebroadcast of AJDCT are higher than JDCT but lower than SB.

6. CONCLUSION

A simple distributed broadcasting algorithm in ad hoc based on the algorithm has been proposed and its accuracy has been shown by simulation. This algorithm is

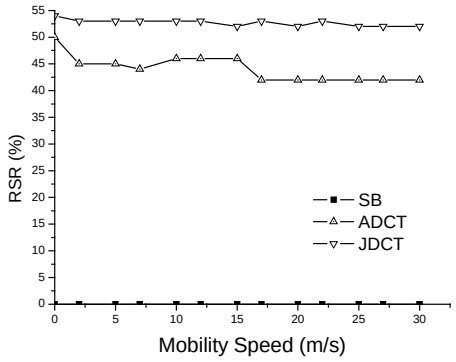


Fig. 5. RSR vs. Mobility speed

based on adaptive changed distance threshold and counter threshold scheme. It runs in a distributed manner by each node without needing any global information. It gets a higher broadcast coverage than JDCT at the cost of adding a little protocol overhead. The simulation experiments have demonstrated the efficiency of proposed broadcast algorithm. It's suitable for mobile ad hoc networks.

7. REFERNCES

[1] L.Y. Li, CH.L. Li, "Genetic Algorithm-Based QoS Multicast Routing for Uncertainty in Network Parameters", Proc. of 5th Asia-Pacific Web Conference, APWeb 2003, Xian, China, Appril, 2003, 23(25): 430-441.

[2] L.Y. Li, CH.L. Li, "A routing protocol for dynamic and large computer networks with clustering topology ",Computer Communications, Elsevier, UK ,2000, 23(2):171-176.

[3] D. B. Johnson, D.A. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", http: //www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt , April 2003.

[4] L.Y. Li, CH.L. Li, Acta Informatica. "A distributed QoS-aware multicast routing protocol ", Computer Science, Springer-Verlag GmbH ,2003, 40 (3):221-233.

[5] J. Cartigny, D. Simplot, "Border Node Retransmission Based Probabilistic Broadcast Protocols in Ad-Hoc Networks", In Proc. 36th International Hawaii International Conference on System Sciences (HICSS'03), Hawaii, USA, 2003.

[6] L.Y. Li, CH.L. Li, "A QoS multicast routing protocol for dynamic group topology", Information Sciences, Elsevier, UK, 2005, 169(1-2): 113-130.

[7] C. E. Perkins, E.M. Royer, "Ad hoc on-demand distance vector (AODV) routing", In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999.

[8] S. Ramanathan, M. Streenstrup, "A Survey of Routing Techniques for Mobile Communication Networks", Mobile Networks and Applications, 1996, 89-104.

[9] E. Royer, C-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", IEEE Personal Communications, IEEE Communications Society, 1999, 6(4): 46-55.

[10] J. Broch, "A performance comparison of multi-hop wireless ad hoc network routing protocols", Proc. ACM MOBICOM, ACM Press New York, 1998,85-97.

[11] Y. C. Tseng, S.Y. Ni, et al "The Broadcast Storm Problem in a Mobile Ad Hoc Network", MOBICOM'99, ACM Press New York, 1999,151-152.

[12] A. Qayyum, L. Vienno, et al "Multipoint relaying for flooding broadcast messages in mobile wireless networks", In Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), Hawaii, 2002.

[13] G. Calinescu, I.I. Mandoiu, et al "Selecting forwarding neighbors in wireless ad hoc networks", Mobile Networks & Applications, ACM Press, 2004, 9(2): 101-111.

[14] ITSI STC-RES10 Committee. Radio Equipment and Systems: High Performance Radio Local Area Network Type 1, Functional Specifications. 1999, 7

[15] E. Pagnani, G. P, Rossi: "Providing reliable and fault tolerant broadcast delivery in mobile ad-hoc networks", Mobile Networks and Applications, DBLP, 1999, 5(4): 175-192.

[16] M. Gerla, TJ Kwon, et al: "On demand routing in large ad hoc wireless networks with passive clustering", Proceedings of the IEEE WCNC 2000, Chicago, IL, September 2000.

[17] W. Li, H. Li: "A dominating-set-based routing scheme in ad hoc wireless networks", Telecommunication Systems, DBLP, 2001, 18(1-3): 13-36.

[18] Ivan Stojmenovic, Mahtab Seddigh et al "Dominating Sets and Neighbor Elimination-Based Broadcast Algorithms in Wireless Networks", IEEE Transactions on Parallel and Distributed Systems, IEEE Computer Society, 2002,1(13): 14-25.

[19] J. Wu, F. Dai: "A Generic Distributed Broadcast Scheme in Ad Hoc Wireless Networks", IEEE TRANSACTIONS ON COMPUTERS, IEEE Computer Society, 2004, 10 (53): 1343-1354.

[20] B. Chen, K. Jamieson et al "An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Network", ACM Wireless Networks, ACM Press, 2002, 5 (8): 481-494.

[21] H. Lim, C. Kim: "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks", In Proc. of the ACM Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), 2000, 61-68.

[22] R. Dube, C. D. Rais, et al "Signal stability based adaptive routing (SSA) for ad hoc mobile networks", Technical Report CS-TR-3646. University of Maryland, College Park, 1996.

[23] A. Vargas: OMNET++Discrete Event Simulation System. version 3.0 edition, 2005.

[24] R. Dube, C. D. Rais, et al, "Signal stability based adaptive routing (SSA) for ad hoc mobile networks", Technical Report CS-TR-3646. University of Maryland, College Park, 1996.

[25] L.Y. Li, Q. Sun, “An Efficient Distributed Broadcasting Algorithm for Ad Hoc Networks”, APPT, LNCS 3756, 2005, 363-372.

Application of Distributed and Parallel Computing in Traffic Network Simulation

Zhicai Juan
Institute of Transportation Studies, Shanghai Jiao Tong University
Shanghai, China
Email: zcjuan@sjtu.edu.cn
Linjie Gao
Institute of Transportation Studies, Shanghai Jiao Tong University
Shanghai, China
Email: lj.gao@163.com
Hongfei Jia
Transportation College, Jilin University
Changchun, Jilin, China
Email: jhfei@email.jlu.edu.cn

ABSTRACT

Traffic network simulation describes movement behaviors and track of individual vehicles. When network is scale-up, the process takes more time and computing resources. Moreover, computing capacity of single processor computer cannot satisfy simulation requirement. A distributed and parallel computing approach to such a large-scale traffic system might be economical and efficient. This paper designs framework of parallel traffic simulation system by using domain decomposition method according to parallel characteristics of traffic network simulation, in other words, traffic network is partitioned to several sub-networks, and each processor of clusters is responsible for a different sub-network of the simulated region. A network-partition algorithm based on the number of vehicle is used to optimize load balancing. Information exchange mechanism between sub-networks is described. The developed parallel simulation system is implemented in parallel computing platform based on Message Passing Interface. From the results of numerical example, the proposed parallel simulation algorithm can improve speed and efficiency of traffic network simulation. This algorithm provides a foundation for larger scale traffic network simulation and real-time applications.

Keywords: Traffic Network Simulation, Distributed and Parallel Computing, Domain Decomposition, Network-Partition Algorithm.

1. INTRODUCTION

In recent years, parallel computing is applied in many fields, such as quantum chemistry, astrophysics, meteorology etc. It is exigent to deal with many complex scientific problems with powerful parallel computers. Application of parallel computing makes computation performances attain obvious improvement. Microscopic traffic network simulation is to simulate the behavior of individual vehicle running in the whole traffic network, analyze the movement rule of system entity and assess performance of system for helping in decision-making and system designing. Traffic simulation system possesses complex entities and property. When the scale of traffic network becomes larger, the speed of traffic network simulation will slow down and simulation computing also needs more time and resources. Temporally, it is hard to achieve computation capacity with a single CPU. Because of the mainframe computer's high price, it is not widely used. However, the distributed cluster has many advantages, such as autonomy, parallel, expansibility, modularization, low cost, and so on. Moreover, it can also accelerate simulation speed and improve computing efficiency. Therefore, it is an effective measure to enhance simulation speed and efficiency and meet traffic real-time management and control in traffic network simulation.

Parallel implementation of traffic network simulation is a relatively new research field. From the 1990s, with the quick development of computer science, some academic research institutes in developed countries such as USA, UK and Germany, have been trying to use mainframe parallel computers or networks to make parallel simulation research on large-scale street networks, and achieved some theoretic and practical fruits. American developed parallel versions of traffic simulation software—TRANSIMS by using cellular automata model. British Quad Stone Company and Canadian SOFTIMAGE Company cooperatively developed PARAMICS simulation software with parallel computing theory, and this software is applied in intelligence transportation systems. Germanic PTV Vision Company and Siemens Company are cooperatively developing VISUM-Online parallel simulation systems. There are more articles about special mainframe computer's parallel simulation and fewer articles about distributed cluster systems' parallel simulation in present bibliography. This paper describes traffic network simulation with distributed parallel computing method [1,2].

2. DISTRIBUTED PARALLEL TRAFFIC NETWORK SIMULATION SYSTEM ARCHITECTURE

2.1 Parallel Characteristics of Traffic Network Simulation

The changes of transportation system states, caused by a lot of events that happened at the same time or interval, are concurrent and parallel. This system belongs to parallel discrete event system. At any time, traffic states change in a region of street network cannot immediately affect other regions' traffics [3]. For example, the behavior of running vehicles in one link cannot immediately affect vehicles in another link. This temporal and special independence can make traffic simulation system parallel. Furthermore, the actual traffic system mostly contains three parts: traffic

network, network control and vehicle. In this system, most activities are parallel in essence, such as vehicle movement and signal states change, etc. Therefore, many entities can be simulated simultaneously in traffic system.

2.2 Distributed Parallel Simulation System Architecture

According to parallel characteristics of traffic network simulation and weak relativity of different simulation area computing, domain decomposition method, which divides computational load into some approximately equal parts, is used in traffic network. An individual PC of the cluster system processes each part. At last, the total simulation results are generated. In simulation process, partial vehicles need to exchange boundary traffic flow information of the neighbor subdomains because they can leave a sub-network and enter into another sub-network on the way to their destinations. And this can keep different simulation regions consistent.

In simulation process, parallel program uses master-slave architecture to equally allocate computational load to each processor of cluster system. Having the knowledge of the overall work to be done, the master process controls general scheduling, spawn slave processes, partition network, assign traffic flow, designate tasks to each slave process, and output simulation results. The slave processes are in charge of simulation computing of each region [4]. The architecture of master-slave parallel simulation is shown in Fig. 1.

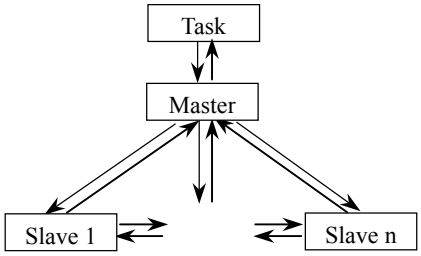


Fig. 1. The architecture of master-slave parallel simulation

The basic idea of distributed parallel traffic network simulation is described as following: Firstly, the master process and the slave processes read entire network data from MS SQL database, divide it into several sub-networks, and allocate the sub-network to each slave process by network-partition algorithm. Then each process is responsible for vehicle running process of one sub-network. All slave processes send the updated vehicle and signal data to the master process, and the master process sends these data to client to show animation and compute running measures of entire network. These two processes need to exchange vehicle information and boundary area information when vehicle traverses the boundary of sub-networks. Clock synchronization needs to be carried out in each simulation step. Traffic network simulation system includes main input module, network control module and microscopic simulation module. Along with the progress of system clock, these three modules cooperatively complete vehicle load process. Meanwhile, in microscopic traffic simulation course, network control module executes synchronously.

According to the idea of distributed parallel traffic network simulation, system architecture is designed as follows:

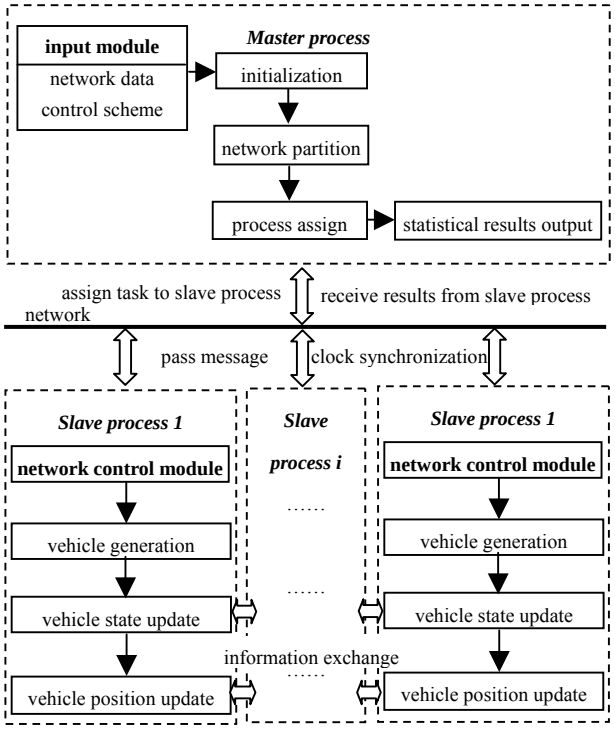


Fig. 2. The architecture of distributed parallel traffic network simulation system

In a parallel environment, information exchange among different areas has two main approaches, which include shared-address space and message passing. In shared-address space approach, all processors have a uniform address space. The variables are kept in a common pool. Therefore they are globally available to all processors. Since only one processor can access the shared memory location each time, increased processor number will lead to severe bottlenecks. In message passing approach, each processor has a private local memory to keep the variables and data. If information exchange is needed between processors, the processors communicate and synchronize by passing messages, which are simple send and receive instructions. With this method, the number of processors is not restricted. This can increase the speed and efficiency of the massive parallel simulation.

MPI stands for Message Passing Interface. It is one of the most important parallel programming tools, which are based on message passing at present. It has many advantages that include favorable transplant trait, powerful function and high efficiency. Meanwhile, it has various free implementation release versions and almost all computer vendors support it. In this respect, this is the main advantage compared with other programming tools. MPI provides luxuriant and efficient communication functions for program designing for high performance on both massive parallel machines and on workstation clusters. It also defines an interface for Fortran and C/C++. MPI supports many communication modes, supplies buffer management functions and safe data transmission, and implements complete asynchronous communication. MPI has abundant data types, including data types of discontinuous space. Furthermore, it also supplies abundant cluster communication functions [5]. Because MPI has powerful communication functions, this paper uses MPI parallel

platform and C++ programming to develop traffic network parallel simulation system in Linux operating system.

3. NETWORK-PARTITION ALGORITHM

In parallel traffic network simulation, to keep load balancing of each processor and minimize communication spending, network-partition is a very important step. According to network-topology architectures, traffic network is decomposed into several sub-networks of similar size. And each processor of parallel computer carries out simulation procedure for one of these sub-networks, which exchanges vehicle data and information among these sub-networks to complete parallel simulation of entire traffic network. Traffic network is cut in the middle of links rather than at the intersections. This separates the traffic complexity at the intersections from the complexity caused by parallelization and makes optimization of computational speed easier. In the implementation, each divided link is fully represented in both processors connected by the divided link. Each processor is responsible for one half of the link while simulating.

After network-partition, sub-networks are of similar size, which means similar computational effort rather than similar geographical area for each sub-network. Because traffic network simulation describes running behaviors and path selecting behaviors of each vehicle, and each vehicle has the same computational effort, the entire computational effort of each sub-network are the sum of all vehicle contained. To have similar computational effort of each sub-network, i.e. each sub-network includes the same number of vehicles, network-partition algorithm that is based on vehicles was proposed to ensure load balancing within processors and minimize communication spending in this paper.

3.1 Basic Idea of Network-Partition Algorithm

To reduce the complexity of traffic behavior and communication spending, intersection is regarded as the basic unit in parallel simulation. Each intersection contains half of all exit and entrance crossing links. Its computational load includes all vehicles located in intersection and links. According to the total computational load of traffic network and the number of processes to be divided, traffic network is divided into some approximately equal parts. Each processor is responsible for one part. Partition position is preferably assigned less computational load link to reduce communication spending. Intersection is the smallest parallel computing unit. In other words, all vehicles located in an intersection are in a sub-network. Furthermore, the intersections located in same sub-network should be linked to each other to decrease communication spending among sub-networks. Because traffic is unloaded in initialized network partition, computational load is divided according to the number of intersections in sub-network partition. Each processor has the equivalent number of intersections. Network partition is run once again according to the number of vehicles of traffic network at regular time interval so as to achieve dynamic load balance.

First, notations used in network partition algorithm are defined as follows:

M_k: the number of intersections in processor k;
M: the total number of intersections in traffic network, $M=\sum M_k$;
N_i: the number of vehicles in intersection i;
N: the total number of vehicles in traffic network, $N=\sum N_i$;
Avg: the average number of vehicles in each processor;
G_k: the number of vehicles in processor k;
$B\ (i, j)$: the adjacency directory matrix in traffic network;
Q: the set of selected nodes in network-partition process;
W: the set of unselected nodes in network-partition process;
S_k: the set of intersections in processor k;
np: the number of processors;

3.2 Network-Partition Algorithm Based on the Number of Vehicles

The particular processes of network-partition algorithm are described as follows:

1) Initialization
   Initializing network adjacency directory matrix $B\ (i, j)$.
   If it is the first time interval of initializing simulation, then go to *step 2*. Otherwise, go to *step 3*.
2) Initializing network partition
   Computing the number of intersections in every slave process.
   If k is less than or equals $M\%(np-1)$, $M_k=M/(np-1)+1$.
   Otherwise, $M_k=M/(np-1)$, $k=1, 2, \ldots, np-1$.
   Selected set of nodes $Q\leftarrow\varphi$, $S_k\leftarrow\varphi$.
   Unselected set of nodes $W\leftarrow \{i\}\ i=1,2\ldots n$.
   $k\leftarrow 1$, selecting a boundary node i, $S_k\leftarrow S_k\cup i$, $Q\leftarrow i$, $i\notin W$.
   While (the number of intersections set S_k is less than M_k)
      Finding adjoining unselected minimized index node j from the adjacency directory matrix of node i. $S_k\leftarrow S_k\cup j$, $Q\leftarrow j, j\notin W$; …
   done;
3) Network-partition based on the number of vehicles
   Selected set of nodes $Q\leftarrow\varphi$, unselected set of nodes $W\leftarrow\{i\}\ i=1,2,\ldots,n$.
   The number of vehicles handled by process k, $G_k\leftarrow\varphi$, the set of intersections, $S_k\leftarrow\varphi$, $k=1,2,\ldots, np-1$.
   Computing all intersections' vehicles number N_i and total network vehicles number N.
   Computing the average vehicles number of each process, $avg=N/(np-1)$.
   $k\leftarrow 1$, Selecting node i maximized N_i from the set of intersections.
   While ($G_k<Avg$)
      Finding adjoining unselected node j, which N_j+G_k is the most close to Avg from the adjacency directory matrix of node i. $S_k\leftarrow S_k\cup j$, $G_k\leftarrow G_k+N_j$, $Q\leftarrow j, j\notin W$.
   done;
4) stop criterion
   If $W==\varphi$, then stop.
   If it is the first time interval of initialized simulation, go to *step 2*. Otherwise, go to *step 3*.

4. MESSAGE PASSING MECHANISM

In cluster's parallel computing environment, communication among processors is based on message and event. An event can be described as an information package including type, address and data. One or more events can be combined as a message, which contains message type and destination processor. In traffic network simulation process, when the vehicle running on the split link leaves a sub-network and enters into another sub-network, the traffic flow information near the boundary of the neighbor sub-network needs to be

exchanged. The data communication of the boundary mostly contains four parts: Confirm the destination processor which the vehicle will enter; Receive boundary information of neighbor sub-network; Calculate speed and move vehicle forward according to car following model; Delete the vehicle information in original sub-network and add the vehicle information in neighbor sub-network [6].

When the vehicle leaves a sub-network and enters into another sub-network in data communication, vehicles states of boundary need to be updated at the same time. However, because two sub-networks are located in different processors, vehicle updating needs to be calculated in different regions. It is likely to appear that back vehicle is updated prior to front vehicle. In this way, the data synchronization problems between sub-networks need to be solved.

This paper maintains the consistency between sub-networks through building vehicle buffers in two-neighbor sub-networks boundary, which is the first 30 meters of half of the divided link. In every simulation step, each processor needs to send vehicle information about the first 30 meters of half of the split link to the other processor. If no vehicle locates in this region, null message will be sent to the other processor. 30 meters is the interaction range of vehicles on a link. In car following model, because only the nearest front vehicle is considered in ken of a lane, only the vehicle which is the nearest apart from boundary is sent to the other region.

Each processor saves its own split link queue structure and the neighbor processors' queue structure. Boundary information exchange must be processed prior to vehicle state updating. Each processor needs to initialize message passing in proper time. Meanwhile, they are waiting to receive boundary exchange information from neighbor processors.

5. PERFORMANCE EVALUATION OF PARALLEL SIMULATION

Parallel simulation performances are mostly evaluated by computational complexity, which includes time complexity and space complexity. Parallel simulation aims to reduce time complexity as soon as possible. Generally it is implemented by increasing the space complexity, such as the space dimensions and the number of processors. The complexity of parallel simulation program has great difference with sequential program. Many factors need to be considered, such as workload, execution time, number of processors, speed-up, efficiency, extensibility and granularity etc. This article mostly describes parallel algorithm efficiency from the following two aspects: execution time and speed-up.

The execution time is an important index of algorithm evaluation. The execution time of single processor can be used to compute speed-up and efficiency of parallel benefit analysis. The execution time of parallel program is defined as the total time elapsed from the time that the first processor starts execution to the time that the last processor completes execution. The execution time includes computing time, I/O time and communication time.

Speed-up is described as obtained accelerative multiple while solving one problem with multi-processors. We can define speed-up as in the following formula.

$$S_p = \frac{t_{seq}}{t_p}$$

Where, p is the number of processors; t_{seq} is the time of one processor solving the problem; t_p is the time of p processors solving the same problem.

This paper tests parallel simulation benefit by a virtual network. Network consists of 60 intersections. The number of processors in parallel environment is 1, 2, 3, 4, 5, 6 and 7 respectively. Simulation time is 600s, 1200s, 1800s, 2400s, 3000s and 3600s respectively. Configuration is legend terminals (P4/2.4G/256M). Table 1 shows comparison between parallel and serial program execution time. Table 2 shows speed-up under different number of processors. Fig. 3 shows speed-up curve.

Table 1. Comparison between Parallel and Serial Program Execute Time

Simulation time /s	Serial program	Parallel program					
		2	3	4	5	6	7
600	291.4	157.7	125.8	99.2	93.2	77.9	81.3
1200	514.1	300.2	250.7	201.3	194.9	163.2	185.3
1800	760.9	464.6	410.4	335.8	330.4	277.9	314.7
2400	1024.6	650.9	606.8	502.7	481.3	421.0	485.6
3000	1308.0	860.0	811.3	700.4	687.7	586.6	691.0
3600	1608.8	1093.7	1015.4	929.6	887.9	786.6	928.8

Table 2. Speed-up Under Different Number of Processors

Simulation time /s	Speed-up					
	2	3	4	5	6	7
600	1.85	2.32	2.94	3.13	3.74	3.58
1200	1.71	2.05	2.55	2.64	3.15	2.77
1800	1.64	1.85	2.27	2.30	2.74	2.42
2400	1.57	1.69	2.04	2.13	2.43	2.11
3000	1.52	1.61	1.87	1.90	2.23	1.89
3600	1.47	1.58	1.73	1.81	2.05	1.73

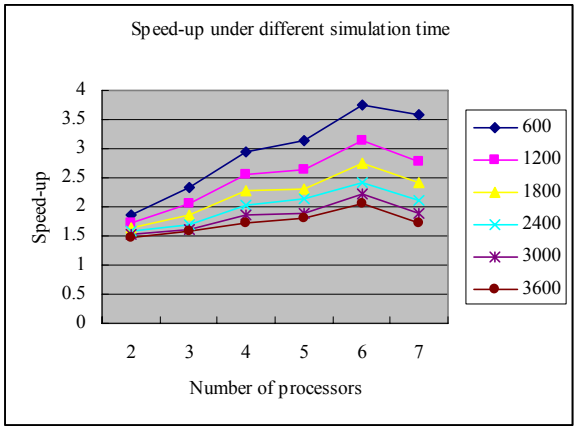


Fig. 3. Speed-up curve

From the above tables and figure, we can learn that distributed parallel computing has been applied successfully in traffic network simulation.

6. CONCLUSION

This paper expatiates on necessity that distributed parallel computing is applied in traffic network simulation and builds distributed parallel simulation system structure. To keep load balance on each processor and obtain least communication spending, network partition algorithm based on the number of vehicles is proposed. In the method, vehicle buffers are built on the divided links and each processor has preferable information exchange and synchronous clock. Eventually, we give an example based on MPI parallel simulation platform and evaluate its

performance.

7. ACKNOWLEDGMENTS

Grant number 50378042, 50338030, and 70371022 from the National Natural Science Foundation of China (NSFC) founds this research. The authors would like to thank NSFC for their support.

8. REFERENCES

[1] T. Junchaya , G.L.Chang, “Exploring real-time traffic simulation with massively parallel computing architecture”, Transportation Research A, 1993, 57-76.

[2] H.X. Liu, W.T.Ma et al, “A distributed modeling framework for large-scale microscopic traffic simulation”, The 84th Annual Meeting of the Transportation Research Board, Washington, D.C, USA, 2004.

[3] L.Y.Wei, “Study on modeling distributed microscopic traffic network simulation”, Dissertation of PH.D, Transportation college, Jilin University, 2002.

[4] L.J.Gao, Z.C.Juan et al, “Design and benefit analysis of distributed and parallel microscopic traffic simulation”, Journal of Highway and Transportation Research and Development, 2006, 4:94-98.

[5] G.L.Chen, H.An et al, Parallel algorithm practice, Beijing: High Education Press, 2004.

[6] G.Qiu, Z.H.Li et al, “A distributed parallel computing structure for traffic simulation”, Central South Highway Engineering, 2004, 1:1-4.

Task Delegation-Based Access Model in Distributed Environment*

Chunhua Gu, Jing Gu, Xueqin Zhang
School of Information Science and Engineering, East China University of Science and Technology
Shanghai, 200237,China
Email: chgu@ecust.edu.cn , gujing01@msn.com, zxq@ecust.edu.cn

ABSTRACT

Virtual enterprise is a kind of distributed environment in which the members need to dynamically collaborate each other. An access model is essential in order to control the access to shared resources among member enterprises in a virtual enterprise. We researched the existing traditional access models, which serve the single enterprise and collaborative environments. By incorporating trust management with layer based access control (LBAC) and task based access control (TBAC), we put forward a Task Delegation-based Access Model (TDBAM). In TDBAM, member enterprises are connected by task assignments. Using delegation logic the inter-organizational control rules are enforced among member enterprises. Thus we can protect the resources of each enterprise in distributed environment efficiently and flexibly. And the needs of access control among enterprises are met.

Keywords: Trust management, Virtual Enterprise, Access model, TDBAM.

1. INTRODUCTION

With the development of society, the form of enterprise's organization changes frequently and virtual organization is a trend. Facing the incorporated world, many enterprises cannot afford the variety of market environments by themselves. Thus a rising form of distributed enterprise's organization -- virtual enterprise [1] is becoming a popular choice among enterprises.

Virtual enterprise is a distributed cooperative platform among many real member enterprises. Each member in a virtual enterprise is independent of other members. In such a distributed environment, it is critical to share resources among the members securely. On the other hand, it is also very important to protect the sensitive information between member enterprises. More and more researchers pay their attention on access control model, an important component of the virtual enterprise's security mechanism.

Our work is devoted to the research of trust management based access control in distributed environment. On the basis of summarizing several existing access models, a Task Delegation-based Access Model (TDBAM) is put forward.

2. BACKGROUND

2.1 Requirement of Applications in Virtual Enterprise

Virtual enterprise has the characteristics of mobility, flexibility and distributed. Once the cooperative purpose is achieved, such kind of enterprise alliance may be disbanded. Therefore virtual enterprise can be provisional or long-term and the participants of a virtual enterprise are dynamic. Virtual enterprise is a kind of enterprise cooperation based on connected informational network.

An application system in a virtual enterprise [2] needs to support the following functions:

1) Protect objects within a member enterprise and between member enterprises. There are two kinds of subjects:(1) those who are in the same member enterprise with the objects;(2) those who belong to the other member enterprises. The protection policy should be different because the trustworthiness of those subjects is different.
2) Support dynamic relations. The members of a virtual enterprise keep on changing dynamically. The AC system should provide a consistent way to handle the up-to-date inter-organizational protection within a virtual enterprise.
3) Support heterogeneous collaborative environments. Each member enterprise has its own AC policy and AC system in its own network environment. The AC system of a virtual enterprise should support different collaborative environments.
4) Support different collaboration modes. Different virtual enterprises have different collaboration modes, which lead to different structures such as linear structure, tree structure and net structure. The AC system should support those different structures.

2.2 Access Model

Access model includes three key elements. They are: Subjects, Objects and Actions. Some Traditional Access Controls (TAC) have been introduced, such as DAC（Discretionary Access Control）, MAC（Mandatory Access Control）, and RBAC（Role Based Access Control）. But these traditional AC models are only suitable for the single real enterprise because all of them require the centric subjects and objects collection.

To meet the different situations and increase the flexibility and capability, many extended AC models have been proposed. The SALSA security architecture [3] is an extension from the centralized AC models. Because task-specific security policy depends on the semantics of the workflow, only can the person with intimate knowledge of the workflow set the security policies for each task. Task-based Authorization Control (TBAC)[4] is a task-oriented model for access control and authorization. In the TBAC paradigm, permissions are revoked and granted just-in-time fashion based on activities and tasks. Layer Based Access Control (LBAC)[5] puts forward a way to exchange access control information depending on process decomposition and task assignment. The inter-enterprise access control is implemented by updating the local ACL. SecureFlow [6] associates each task in the workflow with Authorization Templates (AT), which is

* This work is supported by Chinese NSF No.60473055.

specified by the workflow designer. According to ATs, authorization is actually granted when the task is started.

2.3 Trust Management

In distributed environment, the resource owner and resource applicant do not know each other, and the grantor does not keep the applicant's identity information. Therefore, access control based on identity is not suitable. In order to carry on authentication, the grantor needs to use the information of the third party who knows the applicant. Usually, the grantor believes in some third parties organization with some depth only in some aspects. These properties make the access control in Internet different from the traditional access control. This kind of technology is called "Trust Management (TM)".

Blaze [7] defines trust management as "a unified approach to specifying and interpreting security policies, credentials, relationships which allows direct authorization of security of security-critical actions". Trust Management Model is the framework that refers to establishing and managing the trust relationship.

Some trust management systems have already been proposed, for example: PolicyMaker, KeyNote, SPKI/SDSI and so on. But in most of these permission-based systems, one cannot describe the fact that the issuer grants permissions to all entities that have a certain property. Li [8] extends well understood logic-programming languages with features needed for distributed authorization and provides a logic-based approach to distributed authorization -- Delegation Logic. And a new kind of trust management language, called "D1LP", was given. In this paper, D1LP is used to express task delegation rules.

3. TASK DELEGATION-BASED ACCESS MODEL

The main idea in TDBAM is to connect members in VE using tasks. Because the immediate cause of virtual enterprise is that one can't accomplish the task by its own. So, it is natural to regard the task as the origin why virtual enterprise exists.

The primary function of TDBAM is to define how an entity joins in VE and how it quits from VE through a certain means. Therefore, we introduced one certifying organization –Source of Authority- called "VECA", which is independence of the each member of VE. The main function of this organization is to construct a public trust in platform among the virtual enterprise members, maintain some basic information of them, and is responsible for managing the level relations between tasks.

Because the operation in TDBAM is distributed and the rules and requests are described by D1LP language, for further more detail about D1LP, please refer [8], we define following sentences.

3.1 Join a virtual enterprise

A Says allowEnter (B, T)
A Says isChildTask (T, S)

"A" is a member of a virtual enterprise VE while "B" is an entity, which has not joined VE or has already joined VE. "T" is a new task that assigned to "B" by "A". S is a direct pre-task "T".

Above two sentences describe such a request: the member of the VE – "A" has a task "S" and "A" hopes to divide out a subtask "T" from "S" and assign "T" to "B". If "B" is not a member of the VE, then by this chance, "B" may become the member of the VE and the contact between "A" and "B" is "T". Members already existed in VE add other entities into VE, which reflects the distribution of TDBAM.

"B" here may be in two kinds of situations: "B" is not the member of the VE or "B" is already the member of the VE. In both situations, the tasks all must be added. This is the key point in TDBAM. The difference lies in whether the information of "B" is known by VECA already. Fig. 3-1 and Fig. 3-2 shows the minor difference.

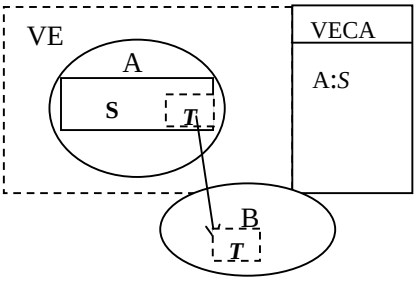


Fig. 3-1. task assigned to new member

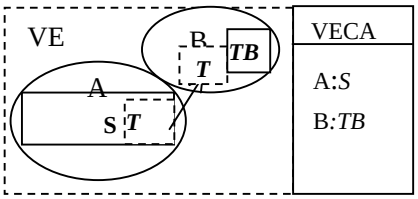


Fig. 3-2. task assigned to exist member.

When VE is set up initially, it includes an initial member, a VECA and a root task of the final goal. Through two above-mentioned sentences, we can add alliance members very conveniently, distributed without any confusion.

In fact, this is really a netted structure. It has broken through the hierarchical structure in LBAC. Between any two members A and B, there may be much task assignment between them. For example, Computer manufacturer A has a task of producing 100 computers. An assigns subtask TA1 (make 50 B brand CPUs) to CPU manufacturer B, and task TA2 (makes 50 C brand CPUs) to CPU manufacturers C. Perhaps CPU manufacturer B, C will require the semiconductor company D to prepare the raw materials (task TB1, TC1 respectively). And, possible C itself is also a semiconductor company too, B may assign task TB2 (prepare materials) to demand to offer materials for C too. Certainly, the distribution of interests among them is not on the consideration range of our paper. Fig. 3-3 shows the above-mentioned situation.

Though the VE's structure is netted, the relation between the tasks is really tree type in TDBAM. TDBAM use the tree relation between the tasks, this is the key that TDBAM can support the netted structure among members.

Adding entities through distributed members makes tasks constructed dynamically in VE of the structure. Initial members do not know where a final task allocation and what the final member's state will be. In other words, subtasks do not need to be set up distinctly at the beginning. Whether the allocation of the subtask happens depends on the members

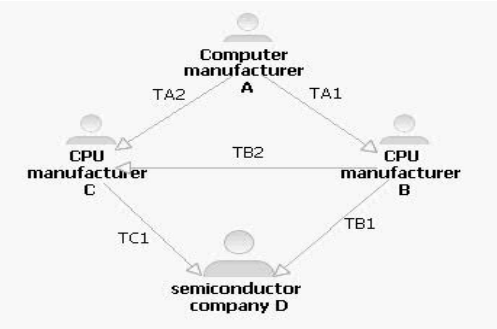


Fig 3-3. task allocation of Netted distribution.

of VE, the upper member of a task (the one who assigned the task) have no ability (right) to instruct down member (the person who accepts task) of the task whether to reallocate the subtask of this task or not. Thus we can improve flexibility of VE in max degree. Each VE member performs its own functions contribute to the improvement of efficiency.

3.2 Withdraw from VE

The situation of withdrawing from VE is a little complicated. Because TDBAM is based on tasks, withdraw from VE is equal with destroying the task relations between members. We defined three kinds of rules for members to withdraw from the VE as following:

A Says remove (B, T)
B Says removeAt (T)
B Says removeAll ()

The meaning of the first sentence "A Says remove (B, T)" is that: Task allocator "A" demands to get rid of the connection about task "T" with allocatee "B", namely we stipulate task allocator has the right to move task relation of task allocatee. But, In TDBAM this kind of remove is not allowed to be acted on tasks sub to "T". Fig. 3-4 has described this kind of situation.

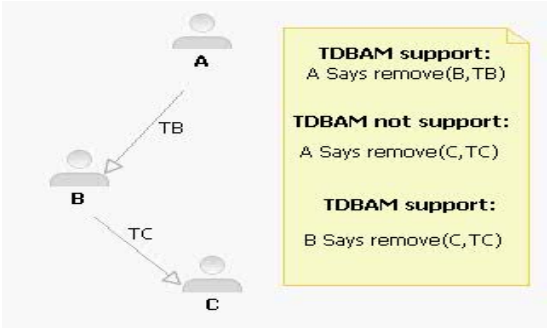


Fig. 3-4. withdraw from VE

Firstly, whether subtask can be assigned depends on members of VE, the upper member of a task (task allocator) have no ability (right) to point out down members (the person who accepts task) of the task whether reallocate subtasks for this task. Though "TC" is the subtask of "TB", and "TB" is assigned by "A", the allocation of task "TC" is nearly free of control of entity enterprise "A", in other words, even "A" might not be clear about whether task "TC" is existing, whether member "C" is existing. So, "A" has no rights of moving "TC" is natural.

Secondly, if VE member "A" can surpass "B" and demand to get rid of C directly, the result will cause unfavorable influence on "B", for example, "B"'s development plan is destroyed. This result is not consistent to the goal of a VE. Any Action that will destroy this goal should all be forbidden.

The meaning of the second sentence "B Says removeAt (T)" is: "B" demands to get rid of task "T". Here "B" is VE members, "T" is the task "B" had. The first sentence is sent out by task-allocator, and this sentence by task allocatee. The application occasions of these two sentences are different. The most possible occasion to use "A Says remove (B, T)" to get rid of "B" is that "A" finds "B" unsuitable to finish task T. "A" perhaps found that the VE member "B" is a bottleneck of the whole project, then determine to break off the task relation with "B". Usually "B Says removeAt (T)" is used when "B" finished task "T" and "B" plans to withdraw from VE after finishing task "T".

Sentence "B Says removeAll ()" is to describe the meaning: VE member B demands to dispel all task relations that it has. In fact, this sentence is an enhancement edition of sentence "B Says removeAt (T)" for breaking off the connection with VE other members completely.

For withdrawing from VE, the elimination of the task relations has transitivity. For example computer manufacturer A assigns the tasks (offers input devices) to the manufacturers B of input device, B distributes the sub task (offers the mouse) to mouse supplier C again. If, for various reasons, A cancels the task (offers the input devices) of B, then the task (offers the mouse) which B assigned to C will be no meaningful, should be got rid of together. Fig. 3-5 shows this kind of situation.

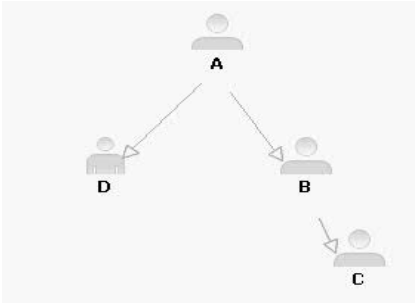


Fig. 3-5. before break away from VE completely

After A Says remove (B, TB), the result is: TB and TC are all dispelled finally.

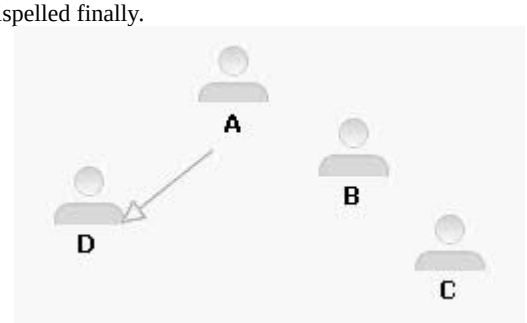


Fig. 3-6. After break away from VE

Another point in withdrawal to VE is: It might not be to cause members of a certain VE to break away from. Because VE members depend on the task as the contact tie, however, TDBAM has not stipulated the quantity of the task relations happened between two VE members. In other words, probably VE member A has a lot of tasks of assigning to VE member B, may even be at the same time VE member B has a lot of tasks to distribute to VE members A. As we see, "A Says remove (B, T)" and "B Says remove At (T)" have one "T" as the parameter in Base Action part, so as to indicate which task is got rid of on earth. Like this,

the result is to cut off one task relation; other tasks remain safe and sound. B remains being the member of the VE.

Obviously, if we want make a VE member completely break away from VE, then it must be completely isolated in VE, namely there has been no any task relations between any other VE member and this VE member. It is that this VE node becomes isolated node. B and C in Fig. 3-6 are such nodes.

3.3 Scope security in TDBAM

Considering the following situation: Computer manufacturer A assigns the task (offers input devices) to the manufacturer B of input devices, B distributes the subtask (offers the mouse) to mouse supplier C again. However, the possibility is, C is not only a mouse supplier, and still a computer manufacturer. So, to computer manufacturer A, C is really still a rival of A. Perhaps A does not hope C can be participated in this serial tasks. However, it was B that invited C to join VE. A is unable to know in advance which enterprises B will assign subtask and give to. Therefore, A might have no idea about B has invited C to join VE. In this way, rights and interests of A virtually are harmed in certain degree, thus is a VE security problem, which we called “the scope security problem”.

TDBAM has considered such a situation and put forward a set of methods in order to remedy the problem caused of the scope security.

First of all, scope security problem cannot be solved with simple method that we refuse one certain entity to join VE in TDBAM. Emphasize one point again here, all origins are tasks in TDBAM, so, it is natural to regard task as the foundation to refuse someone to join VE. Members A may not hope that C appears in this serial tasks of the task (offers input devices), but it may have no special rules on C to appear in other serial tasks. For adding a member carry on dynamically with dynamic distribution of the task in TDBAM, so it is a chance to carry on the legitimacy check when distributing the task. Meanwhile, we offer the expression of asking to forbid someone to be joined:

A Says notEnter (C, T)

“A” is VE members, “T” is a task assigned by “A”. “C” may be the member of VE or not. This regular meaning is, member of the VE, “A”, have already distributed task T to the member of VE, now, A requires that entity enterprise “C” is not allowed to appear in any subtask of “T”. Keep the rule in VECA. Whenever VE members in the task T series distribute the subtask of T, scope security check will be made. When finding disaccord with the scope security rule, TDBAM will stop this action.

Compared with the method refusing entities to be added directly, the grade of the scope security based on task ensured less in term of a certain respect. But, the advantage that it brings is comprehended without being told too, it can offer more flexibility. Make the restriction of the scope security control and stand up more efficient on the basis of the task, can fully utilize the VE members’ resource advantages. See the example of the following:

Hardware manufacturer “A” has assigned task “TAB” (produce CPUs) to give CPU manufacturer “B”, assigned task “TAC” (produce the display cards) to give to the manufacturer “C” of display card. Because of the limitations of production scale, CPU manufacturer “B” assigned subtask “TBD” (produce CPUs) to another CPU manufacturer” D”. But, display card manufacturer “C” may be very strong; at the same time is a CPU manufacturer. CPU manufacturer “B” may fear “C” very much, fear “D” might be cooperative with “C”, so restrained “C” from participating in task “TBD” series. In this case, if adopt the method to refuse directly, then “C” will be unable to continue staying in the VE, then the display card task of “C” is obviously unable to finish, “A” must seek for a display card manufacturer again, thus brought negative influence to operational efficiency of the whole VE. In TDBAM, this problem is very satisfactory and has been solved, “B” has the right to forbid “C” to participate in TBD series, this does not influence “C” to finish task “TAC” of C that “A” distribute, thus has reflected the flexibility.

The restriction of the scope security has succession. This is easy to understand. “B” does not allow “C” to participate in task “TBD”, which means “C” cannot participate in all direct or indirect subtasks of “TBD”. Such a security strategy is necessary; otherwise, TDBAM cannot play any safe function. Imagine if “D” assigned subtask “TDE” of “TBD” to “E” and “E” assigned the subtask of “TDE” to “C” again. If the restriction of the scope security does not have succession, then C will be in the “TBD” series, “B” will unwilling to see that. Here is actually a pair of contradictions of flexibility and security and our TDBAM choose security. If security strategy can't be inherited, then the scope security of TDBAM will seem very narrow; even can say that there is security leak.

4. SYSTEM STRUCTURE OF VE IN TDBAM

We can regard VE based on TDBAM as an abstract network and VE members are just nodes, task relations are just verges of nodes in the whole network, however, be able to make no confusion, we must carry on the division of the systematic system structure to the VE member nodes.

General, we divide VE members’ structures in TDBAM into: Communication interface, D1LP controller, and local strategy unit. As Fig. 4 shows.

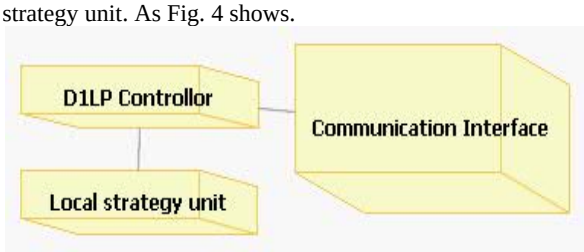


Fig. 4-1. System structure of TDBAM

Communication interface - -This is an essential part. The communication interface can be divided into CA communication interface and member's communication interface according to the difference of targets joined. The main function of CA communication interface is responsible for the communication with VECA, send out message to VECA, request or inquire information of VECA. Member's communication interface is to implement the communication between VE members; it is responsible for accepting D1LP sentence and sending them to D1LP controller and feedback the information of the result. No matter CA communication interface or member's communication interface, each kind of interface has order monitor and report, which are regarded as a VE member's input and output interface respectively. Commonly, order monitor always accept D1LP sentence.

D1LP controller - - This is a key that the whole TDBAM can work normally. We know TDBAM adopts D1LP language to describe the rule with requests and rules between members. The main function of D1LP controller is to calculate D1LP requests received and determine whether according to the rules. Generally, D1LP controller must

include: D1LP Lexical parser, D1LP grammar parser, and D1LP translator.

In the D1LP Lexical parser it should divide D1LP statements into individual vocabulary in order to facilitate the process of the grammar parser.

The function of D1LP grammar parser is to check whether the D1LP statements are meeting the syntax.

D1LP translator is most important. Legal D1LP sentences will be calculated whether they are supported or not.

Local strategy unit. - -It is special and customized place used for preserving D1LP rules for a VE member. Usually it store minimum element for fact (principal), or rules. D1LP controller needs the rule stored in the unit to make "dependence check" in order to judge whether to support the request accepted or not.

Communication interface, D1LP controller, local strategy unit forms the minimum system structure of VE of TDBAM, meanwhile, we think it had better reserve plug-in package interface for expanding need in future.

5. ACCESS STRATEGY IN TDBAM

Because D1LP is adopted as trust management language in TDBAM, each VE member can use D1LP language to make access rules in order to be suitable for itself. We see the example how to use D1LP to implement access control in TDBAM:

Computer manufacturer A assigned task TAB (make CPU) to CPU manufacturers B, assigned task TAC (make display card) to the manufacturer C of display card, assigned task TAD (make CPU) to CPU manufacturers D. Now B needs to offer the information of production of CPU to alliance members, however, B does not want to offer this information to all alliance members, for example, D is CPU manufacturers too, perhaps B does not want D to inquire about this information. Assume that B only want this information inquired by its cooperative partner (mean all entities that assigned task to B), B may make the following access control rules:

B Says readCPUInfo(? m)if VECA Says isPartner(? m,B)

This rule shows, does “B” support any member “? M” operates “readCPUInfo” so long as this member is a cooperative partner of B by VECA authentication.

On the other hand, perhaps CPU manufacturer D has the idea similar to B too. But this time, we assume that D has trust to computer manufacturer A, then, the D may customize the following rule:

D Delegates readCPUInfo(? m)^ 1 to A

It means D delegated “readCPUInfo ()” operate to A with distribution of authority. That is to say, the one A thinks it can carry out “readCPUInfo ()” operate, then D thinks too. In addition, in the example above, it delegate depth of sentence is “1”, then A cannot delegate “readCPUInfo ()” to another member.

See display card manufacturer C, it may only permit A to do some operation, so, the rule can be written as:

C Says getInfo(A).

6. IMPLEMENTATION

Based on the idea of TDBAM, we set up a simulation environment on Windows platform. We mainly focus on basic functions, such as setting-up, the adding of members, calculation of D1LP requests, etc.

Our design follows the characteristics and guidelines of TDBAM. The algorithm of calculate the request is discussed below.

We need to conclude whether support the request or not. And D1LP credentials and rules are the vouchers of making conclusion. We use “request begin” algorithm to process presumption. First of all, we look for the credentials set according to the request to see whether there are credentials can be found. If can be found, it is successful to conclude, conclude is over. If not, we are going to look for the rules set to get the rules which regard D1LP request as its head, send them into a queue. If the queue is empty, it is fail to conclude, conclude is over. Then, we take out rule from the queue sequentially, and find their body part, and then we regard each D1LP sentence of Body part as the new request condition, repeat the step above. Be careful, only when a rule’s all Body sentence is concluded successfully, this rule’s head sentence can be regarded as conclude success.

When the rules have variable in their head or body, our method is to pass the principal to the variable and use the algorithm above. The method which we adopt of support of Delegates sentence is to turn Delegates sentence into “if” rule, this is a kind of simplest treatment way. The difference between them lies in that Delegates statements have clearly delegate depth. What a pity, our D1LP translator has been supported only to the basic grammar characteristic of D1LP, this means, the advanced characteristic such as threshold structure, we have no support.

7. FURTHER DISCUSSION

The migration of the state just means the change of authority. We thought it to be very valuable. It is based on credentials in our TDBAM at present. This has its own advantages, however, if can combine the thought on trust degree, then TDBAM will get powerful expansion. Consider the situation of the following:

Suppose that there is a VE, entity's enterprise A, B, C, D are members. With the description ability of D1LP in TDBAM at present, such a rule we can be very easy to be appointed: So long as two of the three in B, C, D support the operation, A supports too. But, after we introduce the thought on trust degree, situations of the question are different to some extent. First of all, perhaps A has an assessment of trust degree to B, C, D respectively oneself. The simplest, it may be a number value. Then, A may have such a rule to describe: No matter how many of the three in B, C, D, so long as the total trust degree of B, C, D exceed A’s standard, A supports the operate. In fact, to some degree, this situation has been reflected in static unweighted threshold. Now, we can do such expansion even more, the assessment of trust cannot merely be directed against VE members, can also against credentials directly. We still consider the simplest treatment method; we can think that the trust degree of the credential can be inherited on VE members' trust degree. After Introduce the trust degree, we not only look for credentials but calculate the total trust degree of all related credentials as well.

We can protect the resources of each enterprise in virtual enterprise more efficiently, flexible and meet the needs of accessing control among the enterprises through the access control-TDBAM. We are very joyful to see that the research of VE access control has been ripe day and day, this will

further promote the whole development of VE's technology.

8. REFERENCES

[1] E. Lupu, Z. Milosevic,et al, “Use of roles and policies for specifying, and managing a virtual enterprise”, the 9th IEEE International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises, Sydney, Australia, 1999.

[2] D. Hamideh Afsarmanesh, “Management of Distributed Information in Virtual Enterprises”, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands.

[3] M.H. Kang, J. S. Park, et al, “Access control mechanisms for interorganizational workflow”, In Sixth ACM Symposium on Access Control Models and Technologies, 2001.

[4] R.K. Thomas, R.S. Sandhu, “Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management”, R. K. Thomas, R. S. Sandhu Research Associates Cornell Business and Technology Park, 33 Thornwood Drive, Suite 500 Ithaca, NY 14850-1250, USA.

[5] Y. Zhang, M. J. Chung, et al, “Layer-Based Access Control Model in the Manufacturing Infrastructure and Design Automation System”, Proc. of International Calendar of Information Science Conferences 2003, Springer, German, 2004, 197-214.

[6] W. Huang, V. Atluri, “Secureflow: A secure web-enabled workflow management system”, ACM Workshop on Role-based Access Control, 1999, 83-94.

[7] Matt Blaze, Joan Feigenbaum, et al, The KeyNote trust-management system, version 2. IETF RFC 2704, September 1999.

[8] N.H.Li, Benjamin N. Grosof, et al, “Delegation Logic: A Logic-based Approach to Distributed Authorization”, ACM Transactions on Information and System Security, volume 6, number 1, pp. 128-171, February 2003.

Object-Oriented Numerical Software Trilinos and Its Application *

Weiting An1,2, Jianwen Cao2, Ni Zhang1,2
1The Graduate University of Chinese Academy of Sciences
2Laboratory of Parallel Computing, Institute of Software, CAS
Beijing, 100080, China
Email: anwt04@gmail.com

ABSTRACT

Trilinos is Object-Oriented numerical software. It is an effort to facilitate the design, development, integration, and ongoing support of mathematical software libraries. In this paper we firstly present an introduction to Trilinos, its linear package Epetra and solver package AztecOO, and then we use Trilinos to solve linear systems from Matrix Market database on a parallel computer. The sparse matrices are partitioned using hMETIS. The linear solver we used is from AztecOO. This hMETIS-Epetra-AztecOO method combines hMETIS, Epetra, and AztecOO together and can be used in many other scientific and engineering applications.

Keywords: numerical software, parallel computing, iterative methods, Object-Oriented technology, data partitioning.

1. INTRODUCTION

The Trilinos Project [1,2,3] is sponsored by the official programs such as ASCI and LDRD, and is being carried out at the Sandia National Laboratories. Trilinos is delivered via the Gnu Lesser General Public License (LGPL).

The goal of the Trilinos Project is to develop parallel solver algorithms and libraries within an object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific applications [1]. Its linear package Epetra [4] is a package of classes for the construction and use of serial and distributed parallel linear algebra objects. Epetra is one of the base packages in Trilinos. Many Trilinos solver packages can use Epetra objects to provide basic linear algebra computations. AztecOO [5, 6] provides an object-oriented interface to the well-known Aztec [7] solver library. Furthermore, it allows flexible construction of matrix and vector arguments via Epetra matrix and vector classes.

The solution of a sparse system of linear equations $Ax=b$ via iterative methods on a parallel computer gives rise to a graph partitioning problem. A key step in each iteration of these methods is the multiplication of a sparse matrix and a dense vector. Partitioning the graph corresponding to the matrix A properly significantly reduces the amount of communication required by this step [8]. Since Trilinos doesn't provide any tools for data partitioning and load balance, and hMETIS [9] is an excellent software package for partitioning unstructured graphs, combining the abilities of hMETIS and Trilinos can be valuable. This paper presents an hMETIS interface to Epetra-AztecOO as well as an algorithm for hMETIS-Epetra-AztcOO method. Testing results show clear performance enhancement, reducing the time required to solve the system of equations.

* Supported by the Major State Basic Research Project of china (No. 2005CB321700).

This paper is organized as follows: in section 2, 3 and 4 we introduce the Trilinos project, Epetra and AztecOO respectively. Data partition and hMETIS will be discussed in section 5. In section 6 we present the process for hMETIS-Epetra-AztecOO method.

2. OBJECT-ORIENTED NUMERICAL SOFTWARE TRILINOS

Trilinos is a collection of compatible software packages that support parallel linear algebra computations, solution of linear, non-linear and eigen systems of equations and related capabilities. Trilinos packages are primarily written in C++ using object-oriented techniques, while they also provide some C and Fortran user interface support. Trilinos uses a two-lever software structure designed around collections of packages. A Trilinos package is an integral unit, usually developed to solve a specific task, by a relatively small group of experts in a particular algorithms area such as algebraic preconditioners, nonlinear solvers, etc. packages exist beneath the Trilinos top level, which provides a common look-and-feel, including configuration, documentation, licensing, and bug-tracking [2]. Each package has its own structure, documentation and set of examples, and it is possibly available independently of Trilinos. However, each package is even more valuable when combined with other Trilinos packages.

Currently, Trilinos has evolved to the 6th edition, and is still in active development. Every version of Trilinos publishes two releases: general release and limited one. The former comprises the more mature packages; while the latter includes packages, which may be changed greatly thereafter. The general Trilinos 6.0 release has 18 packages, and limited release has 26 packages. According to their functionalities, the packages in the general release 6.0 can be classified into eight categories as shown in table 1.

3. BASIC LINEAR ALGEBRA PACKAGE: EPETRA

Epetra is the basic linear algebra package in Trilinos. Epetra provides the fundamental construction routines and services functions that are required for serial and parallel linear algebra libraries.

3.1 Major Epetra features

Epetra is the current production version of Petra Object Model [1]. It is written for real-valued double-precision scalar field data only, and restricts itself to a stable core of the C++ language standard. Epetra combines in a single package (i) support for generic parallel machine descriptions, (ii) extensive use of standard numerical libraries, (iii) use of

Table 1. Main packages of trilinos 6.0 general release

Category	Package	Description
Basic linear algebra package	Epetra	Essential implementation of Petra Object Model
	EpetraExt	A set of extensions to Epetra
	Kokkos	Sparse and dense kernels
Abstract interface	Thyra	Abstract solver APIs and adaptors
Lineal solver	Amesos	Object-Oriented interfaces to 3rd party solvers
	AztecOO	Concrete preconditioned Iterative solvers
	Komplex	Solver suite for complex-valued linear systems
	Pliris	Dense direct solvers
Preconditioners	IFPACK	Parallel algebraic preconditioners
	ML	Multilevel preconditioners
Nonlinear solver	NOX LOCA	Nonlinear solver methods library of continuation algorithms
Eigensolver	Anasazi	An extensible framework for large-scale eigenvalue algorithms
Common tools package	Didasko	The tutorial of Trilinos
	New-Package	Working package prototype
	Teuchos	A collection of tools across all packages
	TriUtils	Utilities for other packages
Others	PyTrilinos	A set of Python wrappers

Object-Oriented C++ programming and (IV) parallel data redistribution [2]. Fig. 1. show briefly the main class hierarchy of Epetra in UML class diagram (the prefix 'Epetra_' to the class name is omitted for simplicity). It is drawn with Microsoft (R) Visio(R).

Epetra provides two of the six fundamental interoperability mechanisms [2] across Trilinos packages: one is that packages accept user matrices and vectors as Epetra objects and the other is that packages can use Epetra internally. The former implies that any package that accepts user data this way immediately becomes accessible to an application that has built its data using Epetra. While the latter points out that, by using Epetra objects internally, a package can in turn use other Trilinos packages to manipulate its own internal objects.

3.2 Epetra Core Classes

Epetra_Comm, Epetra_Map, Epetra_Operator, Epetra_Multi Vector, Epetra_LinearProblem are among the most commonly used classes in Epetra, thus of great importance. The following gives more details about them.

1) Epetra_Comm: Epetra_Comm is a communication interface declared in Epetra, which encapsulates the general information and services needed for other Epetra classes to run on a parallel computer. Currently it supports serial, MPI and prototype hybrid MPI/threaded parallel programming models [10]. An Epetra_Comm object is required for building all Epetra Map objects, which in turn are required for all other Epetra classes.

2) Epetra_Map: Epetra_Map Contains information used to distribute vectors, matrices and other objects on a parallel (or serial) machine. Basically, the class handles the definition of the global number of elements, local number of elements and global numbering of all local elements.

3) Epetra_Operator: Epetra_Operator and Epetra_RowMatrix are declared as interfaces, only specifying the expected operations but not the specific implementation. This is efficient and convenient, in that the users can easily customize the implementation to satisfy their needs, and software package can interoperability at a more abstract level without being concerned about the underlying details of implementation. For example, the preconditioners generally have a clear aim or require special storage and computing schemes. Then it is convenient to define the special preconditioners by deriving from the operator interface, thus having the advantage of a uniform interface of various preconditioners. To facilitate the use of these two interfaces, Epetra provides two high-quality default implementations: Epetra_CrsMatrix and Epetra_VbrMatrix.

4) Epetra_MultiVector: Roughly speaking, a multi-vector is a collection of one or more vectors that have the same length and distribution. Multi-vector can facilitate processing the systems with multiple right-hand-sides, so that communication cost can be saved and locality of memory access can be increased. It is also useful for block algorithms and applications that manage multiple vectors simultaneously [10]. The Epetra_MultiVector class supports the construction and use of real-valued double-precision dense vectors, multi-vectors, and matrices in a distributed memory environment.

5) Epetra_LinearProblem: The Epetra_LinearProblem class is a wrapper that encapsulates the general information needed for solving a linear system of equations $Ax=b$. It requires an Epetra_RowMatrix or an Epetra_Operator object and two multi-vectors x and b .x must have been defined using a map equivalent to the DomainMap of A, while b using a map equivalent to the RangeMap of A. Linear systems may be solved either by iterative methods or by

direct solvers

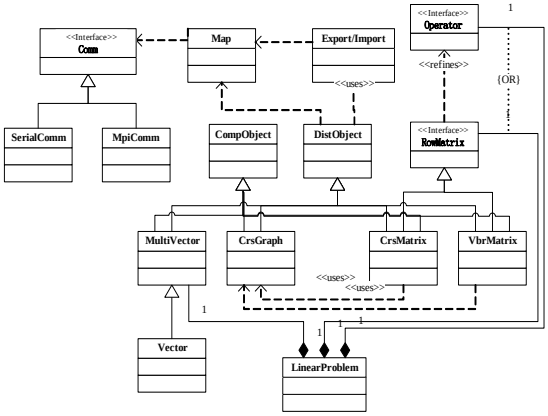


Fig. 1. UML Class diagram: main class in current Epetra

4. LINEAR SOLVER AND AZTECOO

AztecOO is a package within Trilinos that enables the use of the Aztec solver library with Epetra objects. AztecOO provides access to Aztec preconditioners and solvers by implementing the Aztec "matrix-free" interface using Epetra.

4.1 An Iterative Library: Aztec

Aztec is an iterative library that greatly simplifies the parallelization process when solving the linear systems of equations $Ax = b$ where A is a user supplied $n \times n$ sparse matrix, b is a user supplied vector of length n and x is a vector of length n to be computed. Aztec is intended as a software tool for users who want to avoid cumbersome parallel programming details but who have large sparse linear systems, which require an efficiently utilized parallel processing system. A collection of data transformation tools are provided that allow for easy creation of distributed sparse unstructured matrices for parallel solution. Once the distributed matrix is created, computation can be performed on any of the parallel machines running.

Aztec includes a number of Krylov iterative methods such as conjugate gradient (CG), generalized minimum residual (GMRES) and stabilized biconjugate gradient (BiCGSTAB) to solve systems of equations. These Krylov methods are used in conjunction with various preconditioners such as polynomial or domain decomposition methods using LU or incomplete LU factorizations within subdomains.

4.2 An Object-Oriented Follow-On to Aztec: AztecOO

AztecOO is an object-oriented follow-on to Aztec. As such, it has all of the same capabilities as Aztec, but provides a more elegant interface and numerous functionality extensions. AztecOO contains a variety of classes to support the solution of linear systems of equations of the form $Ax = b$ using preconditioned iterative methods.

AztecOO also fully contains Aztec, so any application that is using Aztec can use the AztecOO library in place of Aztec. The primary AztecOO class is of the same name, AztecOO. An AztecOO class instance acts as a manager of Aztec, accepting user data as Epetra objects. If an AztecOO object is instantiated using Epetra objects, all of Aztec's preconditioners and Krylov methods can be applied to the Epetra-defined problem. However, AztecOO provides a variety of mechanisms to override default Aztec capabilities. Users can construct and use preconditioners from Ifpack [10] or ML [10], or can use another instance of the AztecOO class as a preconditioner. Users can also override the default convergence tests in Aztec and use any combination of available status tests in AztecOO status test classes, or define their own.

4.3 Use of Epetra

AztecOO relies on Epetra for both concrete and abstract classes that describe matrix, vector and linear operator objects. Although concrete classes are needed to construct matrices, AztecOO itself uses these matrices via two Epetra abstract classes. By using abstract interfaces, it supports any of the predefined classes that implement the abstract interfaces and allow users to define new implementations. This allows AztecOO to be easily extended.

5. DATA PARTITIONING AND HMETIS

When solving a sparse system of linear equations $Ax=b$ via iterative methods on a parallel computer, here comes a graph partitioning problem. A key step in each iteration of these methods is the multiplication of a sparse matrix and a dense vector. Partitioning the graph corresponding to the matrix A properly significantly reduces the amount of communication required by this step [8]. Since Trilinos doesn't provide any tools for data partitioning and load balance, and hMETIS is an excellent software package for partitioning unstructured graphs, combining the abilities of hMETIS and Trilinos can be valuable.

5.1 Hypergraph and Hypergrahp Models

A hypergraph [11] is a generalization of a graph, where the set of edges is replaced by a set of hyperedges. A hyperedge extends the notion of an edge by allowing more than two vertices to be connected by a hyperedge.

The traditional graph model for decomposing sparse matrices for parallel matrix vector multiplication has two deficiencies. First deficiency is that it can only be used for symmetric square matrices. The second deficiency is the fact that the graph model does not reflect the actual communication requirement. [12] proposed two hypergraph models：column-net model, row-net model. The proposed models avoid all deficiencies of the graph model, enable the representation and hence the decomposition of unsymmetrical square and rectangular matrices as well as symmetric matrices. Furthermore, they introduce a much more accurate representation for the communication requirement.

5.2 METIS and hMETIS

METIS [13] is a family of programs for partitioning unstructured graphs and hypergraphs and computing fill-reducing orderings of sparse matrices. The underlying algorithms used by METIS are based on the state-of-the-art multilevel paradigm that has been shown to produce high quality results and scale to very large problems. The METIS family consists of three different packages: METIS, ParMETIS, hMETIS. METIS is a set of serial programs; ParMETIS is an MPI-based parallel library; hMETIS is a set of programs for partitioning hypergraphs.

The algorithms in hMETIS are based on multilevel

hypergraph partitioning described in [14], and they are an extension of the widely used METIS graph-partitioning package described in [15]. Traditional graph partitioning algorithms compute a partition of a graph by operating directly on the original graph. These algorithms are often too slow and/or produce poor quality partitions. Multilevel partitioning algorithms, on the other hand, take a completely different approach [16]. These algorithms are highly tuned and allow hMETIS to quickly produce high-quality partitions for a large variety of hypergraphs.

hMETIS provides the *shmetis*, *hmetis*, and *khmetis* programs that can be used to partition a hypergraph into k parts. We choose *shmetis* for it is suited for those users who want to use hMETIS without getting into the details of the underlying algorithms. The *shmetis* program is invoked by providing three arguments at the command line as follows:

shmetis HGraphFile Nparts UBfactor

HGraphFile is the name of the file that stores the hypergraph. We refer the reader to [9] for a detailed description of its format. *Nparts* is the number of desired partitions. *UBfactor* is used to specify the allowed imbalance between the partitions during recursive bisection. This is an integer number between 1 and 49.

6. NUMERICAL APPLICATION

We choose testing matrices from Matrix Market database to give an example. These linear systems are solved using hMETIS and Trilinos Software Framework.

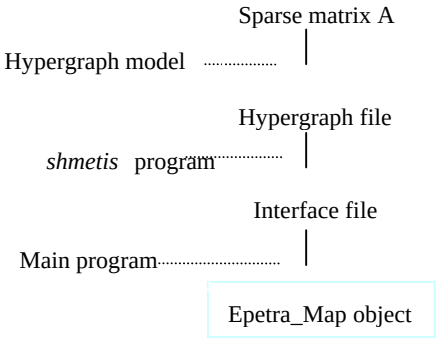


Fig. 2. A hMETIS interface to epetra-AztecOO

6.1 An hMETIS Interface to Epetra-AztecOO

We use the column-net hypergraph model [16] to map sparse-matrix vector multiplication. In this model, matrix *A* is represented as the hypergraph. The vertices and hyperedges correspond to the rows and columns of matrix *A*, respectively. First of all, we generate a hypergraph file and a value file corresponding to a matrix *A* from Matrix Market. The hypergraph file stores the hypergraph corresponding to the matrix *A* using column-net model. The value file lists the nonzero matrix elements one per line in row major order by specifying row index, column index and the value. Secondly, we use the *shmetis* program with the hypergraph file and the number of desired partitions as command line parameters. The following is an example:

prompt% shmetis file.hgr 5 4

file.hgr is the hypergraph file that stores the hypergraph; *5* is the number of desired partitions; *4* are used to spedify the allowed imbalance. The output of *shmetis* program is a partition file. The ith line of the partition file contains the partition number that the ith row of matrix belongs to. The partition file serves as a hMETIS interface file, and will be supplied to hMETIS-Epetra-AztecOO main program as an input parameter from the command line. In the main program, according to the partition information in the interface file, Epetra_Map objects are created. This process is shown in Fig. 2.. Now, we have the distribution of the Matrix across the processors .In the distribution, the number of the rows assigned to each processor is almost the same so the computations is balanced among processors ; the number of the hyperedges cut is minimized, thus the communication among different processors is minimized.

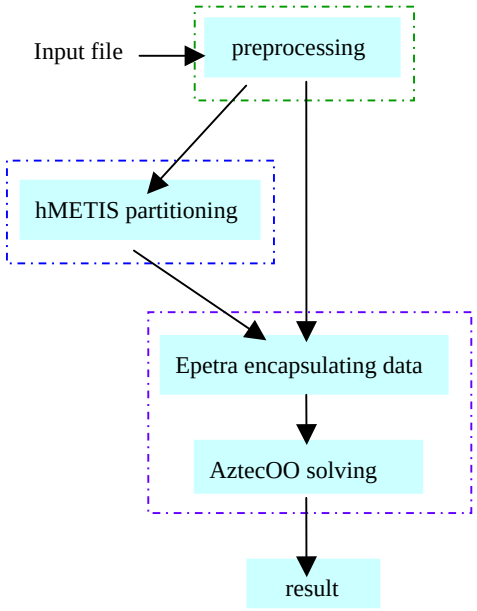


Fig. 3. Program modular structure and WFC

6. 2 Algorithm for hMETIS-Epetra-AztecOO Method

This method accepts two arguments: an interface file and a value file. The interface file is the partition file generated by *shmetis* program. The value file stores the sparse matrix in the previously mentioned format. In main program, Epetra objects are defined based on the information in the interface file and the value file, and then AztecOO solver is called. The primary steps of the method are shown as follows:

1) Define the parallel machine;
2) Open the interface file and the value file;
3) Count the number of rows assigned to each processor; get the global numbering of all local rows;
4) Create an Epetra_Map object;
5) Instantiate an operator *A*;
6) Insert values and indices into *A*;
7) Transform *A*;
8) Create vector *x* and *b* using the same map with *A*;
9) Combine *A*, *x*, and *b* together to define a linear problem;
10) Create an AztecOO instance using the linear problem;
11) Specify solution algorithm;
12) Specify preconditioner;
13) Specify the number of iterations and the residual error, and then solve the linear system,

The work flow of the whole process can be shown in Fig. 3..

7. CONCLUSIONS

In this paper we have presented an overview of Trilinos project, Epetra, and AztecOO. In order to balance computation and minimize communication among different processors When solving a sparse system of linear equations $Ax=b$ via iterative methods on a parallel computer, we give a hMETIS interface to Epetra-AztecOO and combine hMETIS Epetra-AztecOO together. This method is effective to some scientific and engineering applications. Our work is still a primary attempt in this field. Much more has to be done in the future.

8. REFERENCES

[1] M. Heroux, R. Bartlett et al, "An Overview of Trilinos", Tech.Rep.SAND2003-2927, Sandia National Laboratories, 2004.

[2] Michael A. Heroux, Roscoe A. Bartlett et al, "An Overview of the Trilinos Project", ACM Transactions on Mathematical Software, 2004, pp.397-423.

[3] M. N. Phenow, Trilinos home page. http://software.sandia.gov/trilinos/.

[4] M. A. Heroux, Epetra home page. http://software.sandia.gov/Trilinos/packages/Epetra

[5] M. A. Heroux, AztecOO home page. http://software.sandia.gov/trilinos/packages/AztecOO

[6] Michael A. Theroux, "AztecOO User Guide", Tech.Rep.SAND2004-3796, Sandia National Laboratories, August 2005.

[7] R. S. Tuminaro, M. Heroux, et al, "Official Aztec User's Guide: Version 2.1", Tech.Rep.SAND99-8801J, Sandia National Laboratories, November 1999.

[8] Vipin Kumar, A Grama, A Gupta et al, Introduction to parallel computing: design and analysis of algorithms, Redwood: Benjamin/Cummings Publishing Co, Inc, 1994.

[9] G. Karypis, V. Kumar, "hMETIS: A Hypergraph Partitioning Package". http://www-users.cs.umn.edu/~karypis/metis/hmetis

[10] Marzio Sala, Michael A, Heroux, David M. Day, "Trilinos Tutorial version 6.0.X", Tech.Rep.SAND2004-2189, Sandia National Laboratories, September 2004.

[11] G. Karypis, V. Kumar, "Multilevel k-way hypergraph partitioning", Technical Report TR 98-036, Department of Computer Science, University of Minnesota, 1998.

[12] U.V.C atalyurek, C. Aykanat, "Decomposing irregularly sparse matrices for parallel matrix-vector multiplications", Lecture Notes in Computer Science, 1996, pp. 75-86.

[13] George Karypis, Vipin Kumar, METIS home page. http://glaros.dtc.umn.edu/gkhome/views/metis

[14] George Karypis, Rajat Aggarwal, Vipin Kumar et al, "Multilevel hypergraph partitioning: Application in vlsi domain", In Proceedings of the Design and Automation Conference, 1997.

[15] G. Karypis, V. Kumar, "A fast and highly quality multilevel scheme for partitioning irregular graphs", SIAM Journal on Scientific Computing, 1998.

[16] B. Hendrickson, R. Leland. "A multilevel algorithm for partitioning graphs", Technical Report SAND93-1301, Sandia National Laboratories, 1993.

Monte Carlo Distributed Computation For General Distribution Funtion Table Of Probability Distribution Of Statistics

Hengqing Tong, Tianzhen Liu, Yang Liu
Department of Mathematics, Wuhan University of Technology
Wuhan, Hubei Province 430070, P.R.China
Email:hqtong2005@public.whut.edu.cn

ABSTRACT

This paper firstly proposes a new method of generating pseudo-random numbers that enormously breaks through the limitation of pseudo-random number period. Distributed computing is very suitable for this method. Then pseudo-random numbers generated by distributed computing are utilized to construct the distribution function table of statistics that can solve the problem of precision controlling and improving. Finally, this method is applied to calculate distribution function table for computing unit root process test statistics in time series. Meanwhile, a new testing method is proposed to compare with famous Dickey-Fuller test, which have been proved the new method to be better.

Keywords: Monte Carlo Method, Distributed Computing, Distribution Function Table, Dickey-Fuller Test.

1. INTRODUCTION

It is very important to obtain statistics distribution. However, it is often difficult to deduce an accurate distribution in function formula mode for any statistics. Although some distributions can be expressed by subsection functions, most distributions in actual problems cannot be described in this way. Generally, a real distribution can be approximated by its limit distribution. But when the sample capacity is very small, limit distribution is unconvincing. Therefore, we need find a method suitable for general statistics and give out distribution function table of them. Maybe we cannot express the distribution of a statistic in function formula mode, but we can give the function table in any precision. It's a way to solve this problem. This paper has accomplished this method under ordinary circumstances.

Let X be a random variable with a known distribution $F(X)$, and $x_1,\mathrm{L}\ ,x_n$ be its *i.i.d*. samples. The function of the statistic is $y=g(x_1,\mathrm{L}\ ,x_n)$ and g is a known function. We need to give the distribution function table of y. The basic method in this paper is using the Monte Carlo-Distributed Computing algorithm. At first, we take Monte Carlo method to generate a sample $x_1,\mathrm{L}\ ,x_n$. One sample can work out one y that is also a random variable. Repeatedly, we can obtain a n-sample $X_i=(x_1^{(i)},\mathrm{L}\ ,x_n^{(i)})$, $i=1,\mathrm{L}\ ,m$, m independently. Then, we can calculate $y_i=g(X_i)$ $=g(x_1,\mathrm{L}\ ,x_n)$, $i=1,\mathrm{L}\ ,m$. When m is large enough, we can work out the function table of probability distribution of y by using the empirical distribution function of y. The key problem is how to realize sufficient sampling.

The general method to obtain the pseudo-random numbers usually has a restricted period, which cannot satisfy precision request of the distribution function table. Even if we can break through the limitation of pseudo-random number period, generating large enough pseudo-random numbers also needs an extremely huge computation costs.

To solve this problem, a new pseudo-random number combination generator is proposed in this paper. Meanwhile, the distributed computing algorithm helps us to reduce the huge computation costs. Distributed computing can solve a large problem by allotting small parts of the problem to many computers to work and then combining the solutions from the parts into a solution for the problem. Recent distributed computing projects have been designed to use the computers of hundreds and thousands volunteers all over the world via the Internet, such as looking for extra-terrestrial radio signals, looking for prime numbers, finding more effective drugs to fight the AIDS virus and so on. These projects are so large, requiring so much computing costs that they would be impossible for any computer or person to solve in a reasonable amount of time.

At present, some people have unified Monte Carlo method and distributed computing algorithm together to apply in projects with large computation load[1,2]. But there is not a precedent to applying Monte Carlo method and distributed computing algorithm together in calculating the distribution function table of statistics. This paper firstly applies the Monte Carlo method and distributed computing algorithm together in distribution function table calculation. By using distributed computing, we have generated large enough pseudo-random numbers and given out the distribution function table of statistics in any precision.

At last, the way to calculate distribution function table has been used in unit root process testing of the analysis of time series. Then, we propose a new testing method based on that. Usually, Dickey-Fuller test is used to do unit root process testing of time series (example: Examining the American Ministry of Finance bond 1947 to 1989 quarter interest rate). But the Dickey-Fuller test has its limits. Firstly, Dickey-Fuller test uses limit distribution to do testing[3,4]. It tends to have error when meeting a small sample capacity. Secondly, it uses stochastic integral form and has to do quite many statistic transforms. Because of these reasons, there are only a few marginal values of statistics in many works, which always don't meet the actual need. So the two reasons accumulating together cause the practical application effect of this famous testing to be much worse. The new testing method proposed in this paper avoids these defects. It applies easily and has good testing effect. Under the population of distributed computing in local area network, the algorithm of the testing method is easy to complete on computers.

2. GENERATING PSEUDO-RANDOM NUMBERS BY DISTRIBUTED COMPUTING

Using Monte Carlo method to generate the pseudo-random numbers on computers has its defect. The period of the

numbers is affected by the computer word length limitation. When the word length of a computer is k, the period of the pseudo-random numbers with uniform distribution generated by a single congruence generator does not exceed 2^k. When the total number of samples exceeds 2^k, next samples repeat the former samples and the precision of the function table is restricted. So, we need to improve the sampling method. Some method to improve sampling has been proposed by statisticians, such as Y. Wang, M.D. Xiong (2005)[5], Kristian JÄonsson (2005)[6], Michael Mascagni, and Yaohang Li (2004)[7], Billinton R, and Gan L, (1991)[8]. Their ideas are mainly based on the principle of combination and breaking order. By using the combination of some generators, the period of pseudo-random number is longer. But the combination effect is influenced by each generator in it and has limits to the period expanding. This paper uses Monte Carlo-Distributed Computing algorithm to generate the pseudo-random numbers that has good uniformity and independence. Also the pseudo-random number period no longer limits to computer word length. The period is not fixed any longer. We can have free periodical pseudo-random numbers according to real need.

Based on the theory of distributed computing, we use the Client/Server pattern to generate pseudo-random numbers. At first, we produce the pseudo-random number seeds on the Server. Then the seeds are transmitted to the clients via the network. Again the clients generate the pseudo-random numbers by these seeds. Finally, the pseudo-random numbers produced by the clients return to the server through the network. These pseudo-random numbers are processed and transferred by the server. Because the clients are independent and the pseudo-random numbers from the clients are even, the pseudo-random numbers produced by the Client/Server pattern have good independence and uniformity. The concrete procedure to accomplish this method is as follows:

The even distribution pseudo-random numbers z_i, $i = 1,\mathrm{L}\ ,d$ are produced on the server as the seeds. Assigning each seed z_i to a client also means to construct a congruence generator for each seed z_i. The mold of z_i is $u_i = 2^k - a_i$, the period of z_i is T_i, $i = 1,\mathrm{L}\ ,d$ where $a_1 = 0$, $a_2 = 1$, and a_i, $i = 3,\mathrm{L}\ ,d$ are the least integer solutions satisfying the coprime of $2^k - a_1,\mathrm{L}\ ,2^k - a_d$. For example, we take $a_i = 2i - 3$, $i = 2,\mathrm{L}\ ,d$. The order of the d generators is controlled by the sequence of the server transferred from the clients. We randomly select $j = 1,2,\mathrm{L}\ ,d$, and then a random vector $X_{ij} = (x_1^{(i)},\mathrm{L}\ ,x_n^{(i)})$ is obtained. In this way, we can obtain sufficient pseudo-random numbers that can pass every statistical test. Because the seed on each client to produce the pseudo-random numbers is different, so as the mold, the random sequences on various clients don't repeat but are mutually independent. Thus, the period of the total random numbers on the server is $T = \prod_{i=1}^{d} T_i$. Correspondingly, if all the random sequences produced by the clients are of full cycle, the period of the total pseudo-random numbers on the server is $T = \prod_{i=1}^{d} T_i = \prod_{i=1}^{d} 2^k - a_i$. Furthermore, the quantity of the seeds produced by the server rests on the actual situation, so as the clients' number. If having chosen the suitable d, the period of all the pseudo-random numbers can be greatly expanded.

3. MONTE CARLO-DISTRIBUTED COMPUTING ON DISTRIBUTION FUNCTION TABLE

In computing the distribution function table process, the period limit of the pseudo-numbers has already solved by distributed computing algorithm. But it also needs to confirm the significant digit figure, the mathematical precision of the distribution function table and so on. There are many methods to accomplish this, such as expanding the sample capacity, the invariable numeral in the distributed function table confirming as the significant digit, data smoothing, the interpolation, the fitting and so on.

The Problem of Significant Digit of Table

From Glivenko-Cantelli Theorem,

$$p\{\lim_{m\to\infty} \sup_{-\infty<x<+\infty} \left|F_m^*(x) - F(x)\right| = 0\} = 1$$

The convergence is uniform convergence. We take the number m to duplicate, and compare the tables corresponding to m with that corresponding to $2m$. Their identical digits are the significant digits.

Smoothing Data

In order to eliminate the random error and to improve the precision of the table, we take smoothing for the data of the empirical distribution function $F_m^*(y)$. When the nodes are equidistance, the formulae of data smoothing are linear. We can make use of the equidistance of F axis to adjust the fractile. The formulas are given as follows:

The adjusting formulae at the both sides are given as follows:

$$\hat{y}_1 = (31y_1 + 9y_2 - 3y_3 - 5y_4 + 3y_5)/35,$$
$$\hat{y}_2 = (9y_1 + 13y_2 + 12y_3 + 6y_4 - 5y_5)/35,$$
$$\hat{y}_{n-1} = (-5y_{n-4} + 6y_{n-3} + 12y_{n-2} + 13y_{n-1} + 9y_n)/35,$$
$$\hat{y}_n = (3y_{n-4} - 5y_{n-3} - 3y_{n-2} + 9y_{n-1} + 31y_n)/35,$$

We can take smoothing some times until the differences of the data between two smoothing are less than the precision in advance.

Interpolation of the Table

When the data $y_1,\mathrm{L}\ ,y_m$ have been smoothed, we make use of them to construct the continuous empirical distribution function $F_m(y)$ through m nods $(\frac{1}{n}, y_1),\mathrm{L}\ ,(1, y_m)$ and $(0,-\infty)$, $(1,+\infty)$ is continuous on $-\infty < y < +\infty$, $0 \le F \le 1$, and differentiable in $y_1 < y < y_m$, $0 < F < 1$. We take F axis as the argument and to interpolate y-axis so as to obtain the fractile in any probability. Because F axis is equidistance, we can make use of Newton forward or backward interpolation formulae. At

the beginning of table, Newton forward interpolation formula is:

$$f(y) = f(y_0 + sh) = f(y_0) + s\Delta f(y_0) + \frac{s(s-1)}{2}\Delta^2 f(y_0) + \mathrm{L}$$
$$+ \frac{s(s-1)\mathrm{L}\ (s-n+1)}{n!}\Delta^n f(y_0) + R_n(y),$$
$$R_n(y) = \frac{s(s-1)\mathrm{L}\ (s-n)}{(n+1)!}h^{n+1} f^{n+1}(\xi), \quad y_1 < \xi < y_n$$

Where $y = y_0 + sh$, $h = y_i - y_{i-1}$, $i = 1,\mathrm{L}\ ,m$ and

$$\Delta f(y) = f(y+h) - f(y),$$
$$\Delta^2 f(y) = \Delta\Delta f(y) = f(y+2h) - 2f(y+h) + f(y),$$
$$\Delta^n f(y) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f[y + (n-i)h].$$

At the end of table, Newton backward interpolation formula is:

$$f(y) = f(y_n + th) = f(y_n) + t\nabla f(y_n) + \frac{t(t+1)}{2}\nabla^2 f(y_n) + \mathrm{L}$$
$$+ \frac{t(t+1)\mathrm{L}\ (t+n-1)}{n!}\nabla^n f(y_n) + R_n(y),$$
$$R_n(y) = \frac{t(t+1)\mathrm{L}\ (t+n)}{(n+1)!}h^{n+1} f^{n+1}(\xi), \quad y_1 < \xi < y_n$$

Where $y = y_0 + th$, and

$$\nabla f(y) = f(y) - f(y-h),$$
$$\nabla^2 f(y) = \nabla\nabla f(y) = f(y-2h) - 2f(y-h) + f(y),$$
$$\nabla^n f(y) = \sum_{i=0}^{n} (-1)^i \binom{n}{i} f(y - ih).$$

Curve Fitting

The computation result shows that the precision of data in the middle part of table is higher than that at the beginning or at the end of table. However, the data at the beginning and at the end of table are more important. Therefore, we need to make use of the data in the middle part to improve the precision of data at the beginning and at the end of table. In general, a real distribution function is a smooth curve. We can use a smooth curve with subsections to approximate it. It is important to select the number of subsections.

For the nodes $(\frac{1}{n}, y_1), \mathrm{L}\ , (1, y_m)$, we take three piecewise to fit them. Similarly, we take F axis as the argument. Using the polynomial

$$Y(t) = a_0 + a_1 t + \mathrm{L}\ + a_s t^s, \quad (s < m)$$

We take $\delta = \sum_{i=1}^{m} [Y(\frac{i}{n}) - y_i]^2$ to the minimum. To obtain the coefficients of polynomial a_i , $i = 1,\mathrm{L}\ ,s$, we need to solve a linear system of equations by Gaussion elimination method. Then we can calculate the fractile $Y(p)$ in the probability p .

Computation Example

Compute of function table of $\chi^2(10)$ distribution. Take $m = 3000$, $n = 10$. First, we generate n standard normal distribution pseudo-random numbers as seeds on the server. Then we assign these seeds to n clients, each client generates $\frac{m}{n}$ pseudo-random numbers. Last, there are m standard normal distribution pseudo-random numbers on the server. Randomly taking 10 numbers of them every time to calculate their quadratic sum, the process is repeated m times. So we obtain the empirical distribution function $F_m^*(x)$. Then we take data smoothing and curve fitting for $F_m{}^*(x)$ to obtain the fractiles of probabilities. We compare the results with known function table of $\chi^2(10)$ distribution in Table 1.

Table 1. Fractile of $\chi^2(10)$ Distribution

n	a-0.005	0.01	0.025	0.05	0.1	0.25
10	2.199	2.576	3.212	3.912	4.821	6.626
10	2.156	2.558	3.247	3.940	4.865	6.737
n	a-0.75	0.90	0.95	0.975	0.99	0.995
10	12.435	15.838	18.290	20.493	23.158	26.192
10	12.549	15.987	18.307	20.483	23.209	26.188

The first line in Table 1 is confidence level. The second line is the corresponding fractiles by our algorithm. And the third line is the accurate function table of $\chi^2(10)$ distribution. We can see our algorithm is successful. The program to calculate the distribution function table of the statistic has been written by Java language. And the key problem of the distributed computing has been solved by the Remote Method Invocation (RMI). All of these programs can be downloaded from the on-line website: http://public.whut.edu.cn/slx/English/index. HTM

4. FRACTILE CALCULATION OF UNIT ROOT PROCESS TESTING

Unit root process testing has faced the problems of distribution function table. One problem is that the precise distribution of statistics cannot be inferred; the other is that the limit distribution of statistics also is the stochastic integral form. Whereas applying the Monte Carlo-Distributed Computing algorithm, we can simulate the distribution function table directly according to the original distributing of the statistic and the random variable. Then we can calculate the quantile or the probability freely according to the actual situation. Supposing the statistic t of the first auto-regression in the steady time series is given as follows:

$$y_t = \rho y_{t-1} + \varepsilon_t \qquad (1)$$

Where $|\rho| > 1$, $\{\varepsilon_t\}$ are $i.i.d.$ $E(\varepsilon_t) = 0$, $D(\varepsilon_t) = \sigma^2 < \infty$. Examining the supposition:

$$H_0 : \rho = \rho_0 \leftrightarrow H_1 : \rho \neq \rho_0$$

Where $\rho_0 < 1$, $t_T = \frac{\hat{\rho} - \rho_0}{\hat{\eta}}$, $\hat{\rho}$ is the least square estimation of ρ , $\hat{\eta}$ is the estimation of standard deviation of $\hat{\rho}$.

When the supposition is $H_0 : \rho = 1$, t_T doesn't obey the t distribution. So we cannot do the testing directly. Meanwhile, Dickey-Fuller takes measures to solve the problem. He carries on some changes to the primitive statistic of the parameter estimation, infers the limit

distribution of the statistic, carries on the stochastic simulation and calculates the distribution function table for users. But in the common reference books, there only list the sample capacity of three scales: 25, 50 and 100. And the degree of freedom is fixedly, obviously quite rough in application. However, by using the Monte Carlo-Distributed Computing algorithm introduced in this article, we can not only avoid the limit distribution and the stochastic integral but also calculate the quantile in any precision.

When supposing $H_0 : \rho = 1$, the least square estimation of ρ is $\hat{\rho}$:

$$\hat{\rho} = \frac{\sum_{t=1}^{T} y_{t-1} y_t}{\sum_{t=1}^{T} y_{t-1}^2} \tag{2}$$

We can work out the distribution of $\hat{\rho}$ by Monte Carlo distributed computing according to the supposition of the model and the estimation formula of $\hat{\rho}$ [8,9]. The procedure is as follows. We take the initial value y_0 of the sequence in the model randomly, for example, we take $y_0 = 1$. Then the initial value it is given to the server to generate the pseudo-random numbers as the seeds of the clients. So the clients can generate the pseudo-random number sequence $\{\varepsilon_t\}$. According to the Eq.(1), we can obtain a sequence $\{y_t\}$ with the capacity T. Therefore we can calculate one $\hat{\rho}$ by the Eq.(2). Repeating this procedure, we can obtain sufficient enough $\hat{\rho}$. They are random and can form a distribution to exam the primitive supposition $H_0 : \rho = 1$ or other suppositions. The concrete procedure of applying the Monte Carlo-Distributed Computing algorithm in the unit root process testing is as Fig. 1 describing:

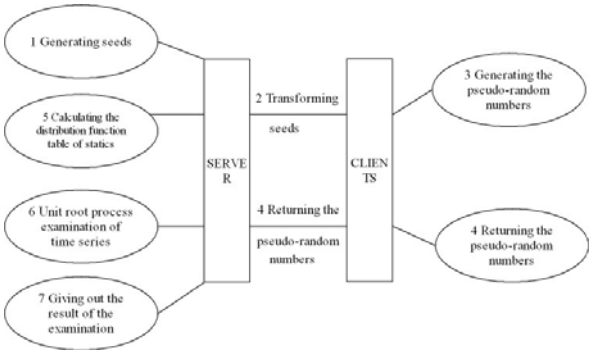


Fig. 1. The Monte Carlo distributed computing of unit root process testing

In applying the mode onto computers, there are several priorities. For instance, we can do the testing of any samples, can reach any precision and shorten the time consuming on computers and so on. Therefore there is important to introduce and apply our algorithm in problems with the distribution function table.

5. REFERENCES

[1] D. Bonacorsi, “Towards the operation of INFN Tier1 for CMS: Lessons learned from CMS Data Challenge (DC04)”, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2006, 559(1): 26-301.

[2] Y. Matsumoto, T. Tokumasu, “Parallel computing of diatomic molecular rarefied gas flows”, Parallel Computing, 1997, 23(9): 1249-1260.

[3] R. Valkanov, “Functional Central Limit Theorem approximations and the distribution of the Dickey-Fuller test with strongly heteroskedastic data”, Economics Letters, 8March 2005, 6(3): 427-433.

[4] A. S. Downes, H. Leon, “Testing for unit roots: An empirical investigation”, Economics Letters, 1987, 24(3): 231-235.

[5] Y. Wang, M.D. Xiong, “Monte Carlo Simulation of LEACH Protocol for Wireless Sensor Networks”, IEEE 10.1109/PDCAT, 2005.

[6] K. JÄonsson, “Using panel data to increase the power of modified unit root tests in the presence of structural breaks”, Applied Mathematics and Computation, December 2005, 171(2): 832-842.

[7] M. Mascagni, Y.H. Li, “Computational Infrastructure for Parallel, Distributed, and Grid-Based Monte Carlo Computations”, Lecture Notes in Computer Science, 2004, 29(7): 39-52.

[8] E.A. Johnson, “Parallel processing in computational stochastic dynamics”, Probabilistic Engineering Mechanics, 2003, 18(1): 37-60.

[9] A. C. Poulain, “Distributed computing with personal computers”, AICHE journal, 1996, 42(1): 290-294.

A Multiplexing Algorithm for High-Density Linecard in Distributed Router

Xiaohui Zhang, Jiangxing Wu, Fenggen Jia
PLA Institute of Information Engineering
No.7 Jianxue Street, Wenhua Road, Zhengzhou, China
Email: {zxh, wjx, jfg}@mail.ndsc.com.cn

ABSTRACT

In distributed routers, it's very import to enhance the throughput and availability of each routing unit. And high-density linecard is a key issue to satisfy these requirements. In high-density linecard, the path to the forwarding unit is shared by the interfaces, so the traffic of different interfaces should be multiplexed. And with the advance in interface speed and the changing nature of Internet, it is more and more difficult to guarantee the fairness of the multiplexing algorithm in high-density linecard of distributed router. This paper proposed an algorithm named LDRR, which is proofed to be fair and possesses the merit of low complexity. An elaborate experiment is provided in the end and which indicates that LDRR is practical and fair indeed.

Keywords: Distributed Router, LDRR, Fairness, Linecard

1. INTRODUCTION

Router is the key equipment of the Internet, and high performance router is the most import issue when constructing next generation network infrastructure. Of all the router architectures [1], distributed structure is the most appealing one for its high throughput and high availability [2] [3]. Generally speaking, a distributed router is a router which is switch-based and the functions of the forwarding engines are integrated into the interface cards [2].

In 2004, China National Digital Switching System R&D Center (NDSC) declared a modified architecture [4] which is shown in Fig.1.

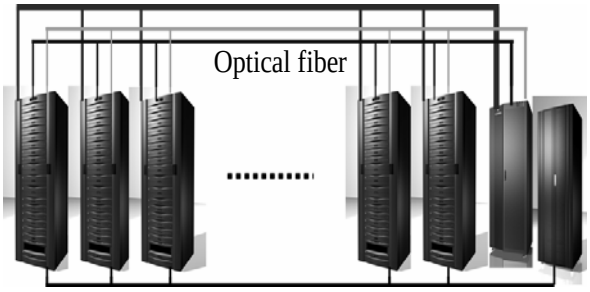


Fig.1 Distributed router architecture

The distributed router in Fig.1 can be shortly described as following. The router is made up of N routing units, and each routing unit has its own routing and switching ability. If a packet's egress and ingress interfaces are in the same unit, then the packet will be forwarded and transmitted within the routing unit. And if the egress and ingress interfaces are in the different routing units, the packet will be switched to other routing unit. The main function modules of each routing unit are linecard, high performance forwarding and port-switch as can be seen in Fig.2. For we are not dedicated to make a detailed description of router architecture, here we only give a further introduction of the high-density linecard.

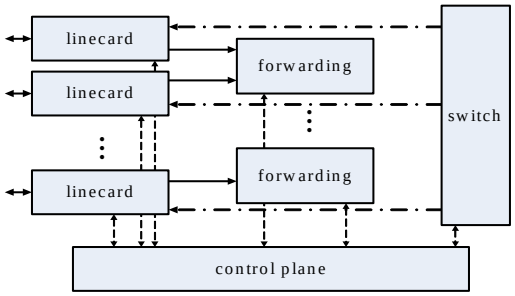


Fig.2 Modules and architecture of routing unit

Nowadays, it has been proved that high-density linecard is very important for high performance routers. The main merits of high-density line include its small-bulk thus to lower the router bulk, multi-interface thus to augment the router's networking ability and it makes it possible to lower the power consumption [3] [5]. It is easily to known that the number of interfaces in a linecard is the number of networks it can accommodates.

An apparent problem emerges once we have a further look into the router and high-density linecard architecture. As we can see from Fig.2, several interfaces in the linecard will share the single path to the forwarding module. Thus it's necessary to multiplex the data (packets) of the interfaces to a single queue. For the complexity of the Internet, i.e. the burst of traffic and other aspects, we should design the multiplexing algorithm carefully to avoid questions described in section II of this paper.

The rest of this paper is organized as follows. Section II discusses the architecture of high-density in distributed router and the problems of multiplexing. Section III describes a multiplexing algorithm named LDRR to resolve the problems. Section IV presents the proof for the fairness of LDRR. Section V provides a testing environment by taking the real characteristics of Internet into account and lists the testing result which is consistent with the analytical result.

2. TRAFFIC MULTIPLEXING

Assuming there are N interfaces in the high-density linecard, then the linecard can be modeled as Fig.3. Each interface in the linecard receives packets from the outside network and queues them. The multiplexer works under the control of the multiplexing algorithm and queues the packets into the single queue for forwarding units.

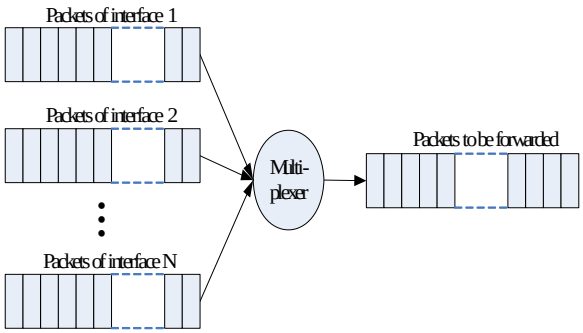


Fig.3: Model of high-density line-card

Works in [6] [7] have shown that Internet packet size distribution is far from uniform and the distributions in different networks may be greatly different. As far as the high-density linecard is concerned, the packet size distributions of the N interfaces may vary greatly. On the other hand, the traffic model is another issue which should be considered. Starting in the early 90's, stimulated by the seminal work [8] of Leland and Willinger, there has been a number of empirical studies that provide evidence of the prevalence of self-similar traffic patterns in Internet (see especially in [9] [10]). Self-similar model means the Internet traffic burst is greater than in traditionally believed Poisson model and the traffic pattern in the networks may be greatly unlike. When it comes to the high-density linecard, the packet patterns to each of the N queues are diverse. Mainly because the differences lie in packet size distribution and traffic pattern, the multiplexer may be unfair in serving the N interfaces if simple round robin algorithm is adopted. The following example presents a further explanation of this unfairness.

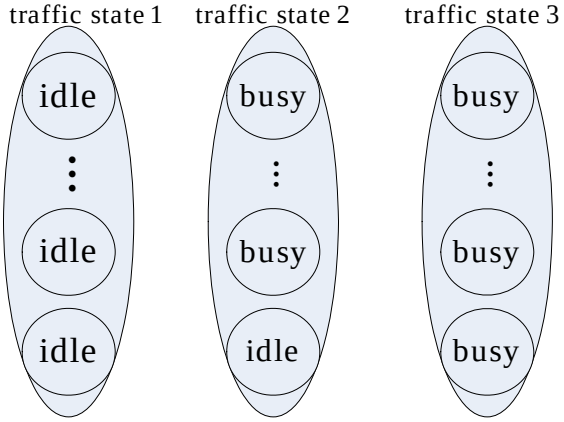


Fig.4: An example of traffic state transition

Consider a scenario of the traffic state transition as shown in Fig.4. At state 1, all the interface traffic is in the state of idle which means no packets received. Then after some time the scenario gets to state2, and at this time the traffic numbered as 1 to $N-1$ is busy and the N th traffic is still idle. When it comes to state3, all the traffic is busy, as we have mentioned in former paragraph, the packet size distribution vary greatly among all the interfaces, so here it may occur that queue 1 to queue $N-1$ each has a long packet to be sent to the forwarding. This may result in that the N th traffic's packets wait too long under round robin, and more severely, the packet queue exceeds the maximal buffer size of the corresponding interface which means big packet loss rate and unfairness. By this example we can see that the multiplexing algorithm is a key scheme in high-density linecard.

When designing the multiplexing algorithm, we can borrow the concepts and ideas from packet scheduling [11] [12]. A number of methods and algorithms [12] have been introduced for packet scheduling, for example, FCFS, PRR, WRR, and Virtual Clock etc. Among all the algorithms, DDR is a very appealing one for its low complexity and good fairness [13] [14]. We adopted the deficit round robin scheme and improved some aspects of the algorithm based on the characteristics of high-density linecard. The newly proposed algorithm is named as LDRR, which is the abbreviation of *deficit round robin suitable for high-density linecard*.

3. LDRR ALGORITHM

The algorithm can be described as following:

i) Every time an interface gets the chance to be served, a whole packet will be transmitted;

ii) The multiplexer assign s_i bytes to the i th interface during each round, and c_i is proportional to it's bandwidth, the counter of that interface is $C_i (i = 1, \ldots, N)$. (a) At the beginning, all $C_i = 0$; (b) When it gets to interface i, change the counter value as $C_i = C_i + s_i$; (c) if the i th interface has packets and $C_i \geq L_{\min}$ (here $L_{\min}$ is the minimal packet length), then the packet (whose length is L_i^1) at the head of the queue will be transmitted, and then change the counter value as $C_i = C_i - L_i^1$;

iii) (d) If there are no packets in the i th queue after step (b), then it goes to serve interface $i+1$, otherwise, repeat (b) until there are no packets remained or $C_i < L_{\min}$; (e) When it comes to i th interface but the packet queue is empty, the only work is set $C_i = C_i + s_i$ and goes to next interface; (f) When the queue is empty and $C_i > C_{\max}$ (here $C_{\max}$ is the maximal of C_i), set $C_i = L_{vg}$ (L_{avg} is the average packet length of the network.), this is to prevent the counter to overflow.

To be more efficient, a queue named $InuseQueu[N]$ is added to indicate the interfaces currently used, thus the multiplexer only need to query the states of the interfaces in the $InuseQueu$.

We will present some further explanation. In ii), one of the condition to serve interface i is $C_i \geq L_{\min}$, this is resulted from our belief that $C_i = L_{\min}$ is the critical state for interface i to be served. Note that the algorithm does not require $C_i \geq L_i^1$, thus unnecessary wait can be eliminated under the circumstances of no subsequent packet. In iii), when the i th interface does not have packets to be sent, then $C_i = L_{avg}$, this will give the interfaces that have been in idle state opportunities to be served once they have packets to send and thus to compensate the counter value lost in idle period.

4. PERFORMANCE ANALYSIS

As we have pointed out, fairness is the most important issue of the multiplexing algorithm, so we will analyze LDRR in this part to verify whether it is fair or not. As far as engineering is concerned, the property of complexity is also very important. So we provide the complexity of LDRR with a theorem.

4.1. Proof of Fairness

Lemma 1: During time interval (t_1, t_2), provided each time the multiplexer comes to the i th interface, the queue always has packets to be sent and let m be the number of the iterations during (t_1, t_2), then

$$mc_i - l_{\max} \le S_i(t_1, t_2) \le mc_i + l_{\max}.$$

Here, $S_i(t_1, t_2)$ is the serving share interface i gets during (t_1, t_2). All the symbols have the same meaning as in section II.

Proof: Provided $C_i(k)$ is the value of counter i after k rounds, and $B_i(k)$ is the number of bytes sent during the k th round, and $S_i(k)$ is the total number of bytes sent during rounds $1 \sim k$, then,

$$S_i(m) = \sum_{k=1}^{m} B_i(k)$$

And also,

$$B_i(k) + C_i(k) = c_i + C_i(k-1).$$

It has been assumed that in each round interface i has packets to be sent, then the maximal number of bytes sent during the k th round is $c_i + C_i(k-1) + l_{\max}$.

And it's apparent that

$$B_i(k) = c_i + C_i(k-1) - C_i(k).$$

Then,

$$S_i(m) = \sum_{k=1}^{m} B_i(k) = mc_i + C_i(0) - C_i(m).$$

Thus we can get,

$$mc_i - l_{\max} \le S_i(m) \le mc_i + l_{\max}.$$

With the relationship between m and (t_1, t_2), it can be get that

$$mc_i - l_{\max} \le S_i(t_1, t_2) \le mc_i + l_{\max}. \ \square$$

Lemma 2: If the i th interface always has packets to be sent, after T (a long enough period of time), the average serving the i th interface gets in each round will be c_i.

Proof: Provided $T = t_2 - t_1$, from Lemma 1 we can get that

$$(mc_i - l_{\max})/m \le S_i(t_1, t_2)/m \le (mc_i + l_{\max})/m,$$

it is equivalent to

$$c_i - l_{\max}/m \le S_i(t_1, t_2)/m \le c_i + l_{\max}/m.$$

When $T \to \infty$, i.e. $m \to \infty$, we can get

$$\lim(c_i - l_{\max}/m) \to c_i,$$

and

$$\lim(c_i + l_{\max}/m) \to c_i.$$

These two limit formulas mean that

$$S_i(t_1, t_2) \to c_i, \text{ when } T \to \infty. \ \square$$

Lemma 3: The bandwidth share of the i th interface is $c_i / \sum_{j=1}^{N} c_j$, and N is the number of interfaces.

Proof: By Lemma 2, the bandwidth share of i th interface is

$$\frac{kc_i}{\sum_{j=1}^{N} kc_j} = \frac{kc_i}{k\sum_{j=1}^{N} c_j} = c_i / \sum_{j=1}^{N} c_j. \ \square$$

Theorem 1: If c_i is proportional to the i th interface's speed t_i, the bandwidth gotten is proportional to t_i.

Proof: Provided $d_p = xc_p$, and x is a constant value. In Lemma 3, do the substitution of $c_p = d_p / x$, then we can get that interface i get the bandwidth proportion of

$$x_p = t_p / \sum_{j=1}^{N} t_j.$$

Consider another interface q, by the same way we can get that the bandwidth proportion it gotten is $x_q = t_q / \sum_{j=1}^{N} t_j$.

Then it can be easily deduced that

$$\frac{x_p}{x_q} = \frac{t_p}{t_q}. \ \square$$

By the analysis we know that LDRR algorithm can guarantee that the bandwidth allocation among the interfaces is fair.

4.2. Algorithm Complexity

Theorem 2: The complexity of LDRR is $O(1)$.

Proof: As we can see in the working scheme of LDRR, it has the same complexity as DRR, that is $O(1)$. $\square$

5. EXPERIMENT AND TESTING RESULT

5.1. Testing Environment

We applied the LDRR algorithm in the LS linecard of our distributed router in NDSC. LS linecard can accommodate up to 48 interfaces and each interface may be configured to work at OC3 or OC12 or OC48 (at most 4), and this linecard is a distributed point of the distributed router. Based on the router architecture described in section 1, the data path in this testing environment is shown in Fig.5. AX4000[15] is the data generator, it can send concurrent data to the RUT's (router under test) LS linecard, and LS multiplexes the data to the forwarding unit, then the switch determines the packets' egress interface and send them out to the receiver. At last AX4000 receives the packets from LS and compares them with the packets sent.

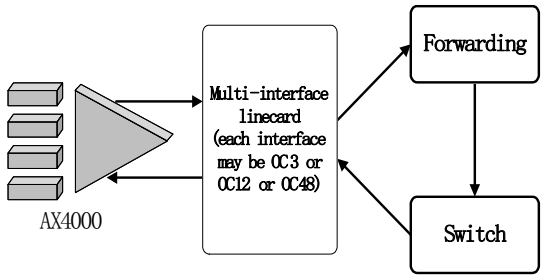


Fig.5: Testing environment

To be consistent with the real Internet environment, we set the testing environment carefully by taking packet size distribution and the traffic model into account.

Packet size Distribution: Currently, the IP packet size distributions used for router test include uniform, Gaussian and multi-modal distribution. To make the test environment more critical, we chose multi-modal distribution. Statistical works have revealed that the packet size distribution has three peak values when the packet size is 64, 576, and 1536. So we carefully chose the tri-modal distribution which follows the peak values just mentioned. Take traffic1 as example, Table1 is the parameter used in tri-modal.

Table1: Parameter for packet size distribution

Parameter	Value	Parameter	Value
Center size1	64	Probability2	30
Half point1	40	Center size3	1536
Probability1	60	Half point3	40
Center size2	576	Probability3	10
Half point2	128	Max length	2000

Traffic model: As we have pointed out that most of the researches on Internet traffic modeling stemmed from the seminal works [8], [9] showing that traffic traces captured on network exhibits LRD property. And recently, researchers find that MMPP may be an accurate and simple model which approximates the LRD characteristics of Internet traffic traces [16], thus we will use MMPP model to generate the testing traffic. The following table is the main parameter used. Take traffic1 as example, the MMPP parameter is shown in Table 2.

Table 2: Parameters for MMPP

Parameter	Value	Parameter	Value
Cross1	0.25	Rate2	17285.12
Cross2	0.3	Rate3	11550.72
Cross3	0.15	Band1	149046.52
Cross4	0.2	Band2	178686.45
Rate1	14,417.92	Band3	119406.58

5.2. Testing result

Table 3: Testing result

Interface Num	Work Mode	Bandwidth Proportion
1	OC3	0.025
2	OC48	0.4
3	OC12	0.1
4	OC3	0.025
5	OC48	0.4
6	OC3	0.025
7	OC3	0.025

In Table 3, we can easily find that each interface get the bandwidth proportion which is linear to its data rate. So we can get the conclusion from this test result that the multiplexing algorithm LDRR used in the linecard is fair.

6. CONCLUSION AND FUTURE WORKS

With the advance of network speed, a challenging problem of guaranteeing the fairness of high-density linecard's multiplexing algorithm emerges in the research community. This paper investigated this problem in detail, and proposed an algorithm named LDRR. Theoretic analysis proofed that LDRR is fair. This paper also provides a case of LDRR's implementation in distributed router to verify its fairness practically. And the testing result is consistent to the analytical results.

In the future, more interfaces with higher speed will be integrated into the high-density linecard, and the Internet is changing all the days, so some parameters in the algorithms may be adjusted with the environment change. And in another way, as we have pointed out that the probability of concurrent busy of all the interfaces is smaller than one and this phenomena is a important clue to lower the requirement for forwarding unit, but no previous works have investigated this. Considering the development of Internet traffic modeling, we will do further work on the packet multiplexing.

7. ACKNOWLEDGEMENT

The authors would sincerely acknowledge Professor Lan Julong and Professor Wang Weiming for their constructive criticisms and suggestions for improving the presentation of this article. And also acknowledge the anonymous DCABES reviewers for their work on this paper. Extensive discussions with Professor Zhang Guojie have been a valuable contribution to this work. Finally, we will acknowledge China High Technology Program (No. 2003AA103510), for the financial and technical support

8. REFERENCES

[1] Aweya J., "IP Router Architectures: An Overview.", Journal of System Architecture, Vol.46, PP483-551, 2000.

[2] H. Jonathan Chao, "Next Generation Routers.", Proceedings of the IEEE, Vol.90, No.9, Step.2002.

[3] Craig Partridge, Philip P. Carvey, "A 50-Gb/s IP Router.", IEEE/ACM Transactions on Networking, Vol.6, No.3, June. 1998.

[4] Wang Binqiang, "Technology Report of Tbit Router.", (Chinese version), 2004.

[5] Isaac Keslassy, "The Load-Balanced Router.", Ph.D Dissertation, Univ. Standford, 2004.

[6] Agilent, "White paper: Testing Next Generation Core Router Performance.", 2005.

[7] Attila Pásztor and Darry Veitch., "The packet size dependence of packet pair like methods.", in Tenth International Workshop on Quality of Service, Miami Beach, USA, May.2002.

[8] Leland, W., Taqqu, M., Willinger, Wilson, "On the self-similar nature of the Ethernet traffic.", IEEE/ACM Transaction on Networking, pp.1-15, Feb 1994.

[9] Paxson, V., Floyd, S., "Wide-Area Traffic: the failure of Poisson modeling." IEEE/ACM Transactions on Networking." pp.226-244, July 1995.

[10] Park, K., Kim, G., Crovella, M., "On the effect and

control of Self-Similar Network Performance.", In SPIE Conf. on Performance and Control of Network Systems, 1997.

[11] S.Golestani, "A self clocked fair queueing scheme for broadband applications.", in Proc. IEEE INFOCOMM'94 1994.

[12] Hui Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Network.", Proceedings of the IEEE, VOL.83 No.10, pp.1374-1395, Oct 1995.

[13] M.Shreedhar, "Efficient fair queuing using deficit round robin.", M.S. Thesis, Dept. of Computer Science, Washington University, 1994.

[14] M.Shreedhar and G.Varghese, "Efficient Fair Queueing using Deficit Round Robin.", In ACM SIGCOMM '95, Aug.1995.

[15] [Online available] http://www.spirentcom.com/

[16] L.Muscariello, M.Mellia, M.Meo, "Markov Models of Internet Traffic and a New Hierarchical MMPP Model", online available at www.sciencedirect.com, 2005.

Dependable Data Computing for Distributed System *

Zhaobin Liu
Department of Computer Science and Technology, Dalian Maritime University
Dalian, Liaoning 116026, P.R.China
Email: zhbliu@newmail.dlmu.edu.cn

ABSTRACT

Data dependability is the prerequisite for other dependability. As we enter a new era of globe collaboration and distributed computing, data storage plays more and more essential role in today's fast-growing network services. However at the same time, data are easily exposed to a wider range of potential security threats. In this paper, we concentrated more attention on dependable data computing in distributed system. We introduce a new dependable storage security system that will automatically select which security storage strategy to use and meet user customized policy goals. The paper gives the descriptions of our architecture and solution, the implementation in Linux kernel level and performance evaluation.

Keywords: Dependable Computing, Distributed System, Trusted Computing, Performance Evaluation.

1. INTRODUCTION

In the early 1970s, Anderson introduced the concept of trusted system [1]. Even after 30 years, Compaq, Hewlett-Packard, IBM, Intel and Microsoft founded the Trusted Computing Platform Alliance (TCPA). The industry consortium focuses on ensuring that all PCs are secure for electronic business transactions, and have announced the release of its first version specification [2, 3]. It provides an open criterion for industry to obtain more dependable computing platform. The information security of computer system mainly includes three aspects: network security, system security and storage security. The research of trusted computing mainly concentrates on the former two aspects. However, due to the excitements of Internet and global economy incorporate, the storage security plays more and more important role in the trusted system. Enterprise information is created, stored and used by various users that depend on its accuracy and instant availability. Stored information can range from financial, legal, and technical to customer-related data. The need to ensure secure access, usage and storage of information is of critical importance since business downtimes or losing information may result in tremendously catastrophe for an enterprise [4]. Obviously, if you can't make your data dependable, then you can't make your computing dependable, either [5].

Over the last years, the traditional storage protection technologies (online techniques using high-density disks, full or incremental tape-based backup, snapshots, remote replication and mirror), are widely deployed. But in distributed system, it is unfortunate that the circumstance is more complicated: heterogeneous storage nodes and servers; how often to fulfill backup; how many copies of same data to preserve; how to schedule the policy of replication; and how to hold the load balance. Furthermore, Different application needs different protection policy. NASD [6] provides a set of measures for storage. SSL has been proposed to protect web traffic, SSH to protect remote terminals, and IPsec to protect Internet traffic more generally [7] and also as the security for iSCSI [8]. Mazieres gives a special mechanism used in the self-certifying file system [9,10]. However, each technique only provides some portion of the protection that is needed. So the bedrock of every technology is relative.

In this paper, we introduce a new dependable data computing for distributed system environment. Instead of asking people to manipulate complicated security operation problem, it can provide appropriate security policy based on the type of application or user customization.

The rest of this paper is structured as follows: section 2 introduces the overview and architecture of our approach. In section 3 we discussed the implementation and solution of our system. Section 4 lists the key aspect of future works and it also includes our conclusions. At last, section 5 gives our acknowledgements.

2. SYSTEM ARCHITECTURE

Distributed storage system often-incorporate heterogeneous collections of servers, switch devices and disk arrays. For instance, there are maybe different storage architecture (RAID, NAS, SAN, etc.), different interconnection topology (Fibre Channel, iSCSI, Infiniband, etc.), different operating system (Linux, UNIX, Windows/NT, etc.), different storage devices (tape, IDE or SCSI disks, JBOD, RAID, etc.). Ignoring the local or partial safety, here we consider all distributed system as whole security architecture to research.

2.1 The Architecture of Our Approach

The resources in the distributed system are usually geographically heterogeneous and geographically distributed. Furthermore, these applications have tasks that need to communicate frequently with each other and may have interdependencies. So the traditional security is not adapted to the distributed computing environment. From Fig. 1., we can see that our system has six main components:

1) Authorization and Authentication
2) User customized requirements
3) Failure descriptions
4) Metadata management
5) Storage descriptions
6) Encryption model

* The work is partially supported by the National Natural Science Foundation of China under Grant No. 60273073 and 60303031.

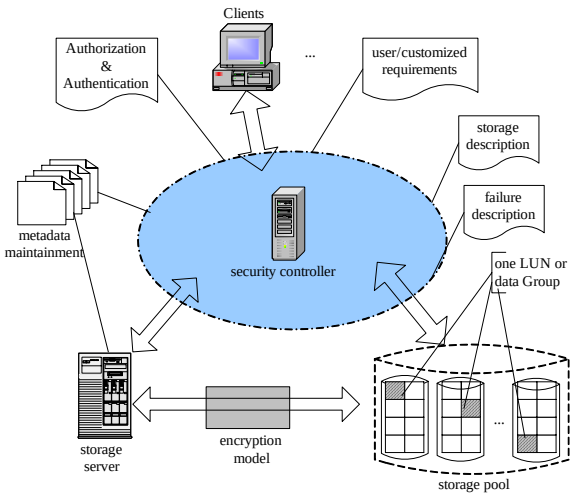


Fig. 1. The hierarchical architecture system

As shown in Fig. 1., we set up a trusted access chain from the upper level application and network layer to storage nodes system. As a result, provide the dependable service for entire system.

2.2 Specification of Host/object in Storage Layer

Trusted computing base (TCB) assign a particular security attribute for his every host and object. Through the attribute comparison of host and object, TCB can control if one host can access some or other object. Due to the different layer of distributed storage system, and accorded with the trusted system concept, we give the specification of host/object in storage system layer (see Table. 1.).

Table 1. Notation of host/object

Layer	Host	Object
Application layer	Clients	Files, directories, mails, etc.
Server Network layer	Request Server	process datagram
Storage device	Server, Other device	Disks, RAID, NAS, etc.
Transport layer	process	link

3. PROPOSED SOLUTION

In this section, we provide a brief overview of our key approach and special implementation: Hierarchical security strategy, authorization and authentication, encryption model, failures/storage description, user-customized specifications.

3.1 Hierarchical Security Strategy

Once the security attribute is assigned, usually it will not be changed by TCB again. In our system, we extended this mechanism seamlessly. We allow security controller change the safety property associated with the customized requirements or state of storage nodes. At the same time, we deploy hierarchical security strategy:

1) Low-level read: in default circumstance, we allow the higher security host can read peer or low-level object.
2) Write Peer: for the sake of data security, we only permit host can write data to the peer object storage nodes.
3) Specified subordinate of host/object: in special instances, some user may need higher grade safety policy. Associated with the user requirement component, user can appoint exclusive object storage nodes or storage devices for some special hosts.

To accomplish the hierarchical security strategy, the security controller should efficiently co-work with other components: examination of a defined security policy to identify the entire subordinate object necessary; management of the functional relationships and interfaces (internal and external); and capturing and analyzing the various specification tables.

3.2 Authorizations and Authentication

One of the most familiar attacks is illegal user to gain access to some data. For instance, after an adversary gains access to data, he could disguise as a storage system file manager, then copy, delete or destroy data discretionarily. So as to prevent unauthorized access, we adopt and enhance the mechanism of Authorization and Authentication.

Every user and node has an exclusive GLOBAL_UID as an identification card. When establishing access session, security controller will check the validity first. On the other hand, after the authorization passed, the system removes the storage server from the data path and allows clients to directly transfer data with the storage devices. We employ centralized authentication that owners delegate responsibility for authentication and authorization to a storage security controller as a trusted third party, which can be easy to manage in a distributed environment. The productivity gains from centralized secure storage administration in a distributed computing environment are self-evident.

3.3 Encryption Model

Based on the effect they have on the data, Erik Riedel summarized three kinds of attacks: leak attacks, change attacks and destroy attacks. To overcome the disadvantage traditional encrypt method, we adopt a new ENCRYPT model based on Stackable file systems [11, 12]. As showed in Fig. 2..

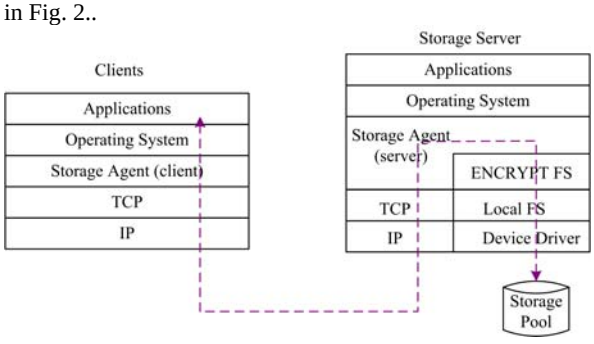


Fig. 2. Design of Encryption model

When a client sends WRITE_Request, the client's storage agent redirects the I/O to storage server. Then the server's agent will call the ENCRYPT file system to encrypt the data and file name, at last, write through the local file system and device driver to storage pool. The read process is the order

reversed. Associated with user demand, the ENCRYPT function can encrypt the file and directory or even whole pathname. For unauthorized user, the data are Perdue or cryptic semantic so as to play down the likelihood of venomous change. At the same time, the data length has not been changed after being encrypted. Obviously, it cannot inflect the global name space architecture and data consistency of distributed storage system.

3.4 Failure Description

In storage system, the attribute of data is very complicated, so it results in the intricate instance for failure and storage specifications.

1) Failure area: Identifying the points of vulnerability and fraud. Organizations must be aware of all the potential points where a security breach might occur.

2) Failure types: the relationship between different failures; which failure type can the technique handle?

3) Failure influence: how often is the same failure? How much scope does the failure cover? (E.g. whole disk array or only one disk, even only some type files).

3.5 Storage Protection Description

The storage protection is essential for clients to select appreciate dependable scheme. For instance, the data is long-lived data or short lived? The data belongs to which LUN or data group? (See Figure 1)

1) Recovery time: how often does the failure occur? How long does recovery take before normal working? If there are deployed fault switch and hot spares, the repair time will be short.

2) Security level: when failures occur, which security schemes the storage system can afford? (E.g. how much data is lost?). How much data can be reconstructed?

3) Fault tolerance: whether the failure can be restore? If can, how long is the failure expected to last? We may be able to estimate the duration of the damage.

4) Storage attributes: when the old disks are retired? When new disk is identified; which scope of the system is affected by the risks?

3.6 User Customized Specifications

1) Identifying failure types: The identification of jeopardy is the prerequisite for user to customize the security strategy. Risks must be identified and described in an understandable way before they can be analyzed. The identifying of failure involves the identification of potential issues, data theft, eavesdropping, fraud, hacker, vulnerabilities, etc. that could negatively affect system running and plans.

2) Prioritizing the attack risks: based on the risks taxonomy and assessments, checklist the quality factor of each risk.

3) Analyzing the probability of failure occurrence and impact degree of data damage

4) Allotting the access authorization: read or write, create or delete, append or truncate, etc.

Failure identification and analysis tools may be used to help identify possible problems. When identifying, prioritizing and analyzing failures, it is good practice to use a standard method for defining and customizing security tactic.

4. PERFORMANCE EVALUATIONS

In this section, we describe the experimental methodology used to compare the performance of ENCRYPT_FS and traditional NFS (without encryption model).

We consider up to 16 clients requests representing light to heavy I/O traffic. Each client reads or writes a 32MB file. We use two metric to compare the performance of NFS and ENCRYPT_FS system. The test results are shown in Table. 2. .The result effect can be found in Fig. 3..

Table 2. Test results

	RAND Read	RAND Write	RAND R/W(50%)
NFS	5.947	4.504	3.954
ENCRYPT_FS	5.472	4.178	3.69
Overhead	7.98%	7.24%	6.68%

We can see that RANDOM Read, RANDOM Write and RANDOM R/W are similar. To the certain test pattern, ENCRYPT_FS mode reduced performance compared with traditional NFS mode. However, the average overhead of ENCRYPT_FS is only about 7%. Actually, the effect is acceptable. One of the most important reasons is that the implementations of ENCRYPT_FS and storage agent localize in the Linux kernel level.

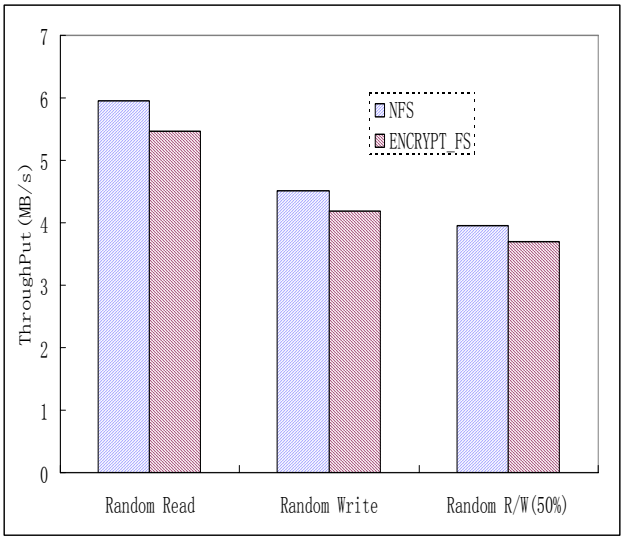


Fig. 3. Throughput comparison with ENCRYPT_FS and NFS

5. CONCLUSIONS AND FUTURE WORK

Getting the high data dependability is a monumental challenge. Most distributed computing environments contain a wide variety of server and storage hardware. In order to efficiently provide strong security on distributed storage system, we bring forward a new security mechanism to consider many aspects of the secure problem. This paper has outlined our architecture and key solutions: hierarchical security strategy, data encryption, proposed specification, etc.

Areas of future work as follows: First, at present, it is difficult to share data, which affiliated with different secure domain. So we expect the occurrence of universal dependa_

ble data standards. Second, ineluctably, the software encryption leads to the overhead of system performance. Of course, we can adopt the more efficient algorithm to encrypt data. However, the granularity of protection is the key influence to the overhead. In general, coarser granularity results in costly expense, whereas, finer granularity leads to a potentially better performance. Finally, no system is dependable enough. We hope to consummate our solution continuously.

6. REFERENCES

[1] Q.P. Guo, Y. Paker, "Optimum Tactics of Parallel Multi-grid Algorithm with Virtual Boundary Forecast Method Running on a Local Network with the PVM Platform", Journal of Computer Science and Technology, July 2000, 15(4), pp.355-359.

[2] J.P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51. Vol. I. AD-758 206. ESD/AFSC, Hanscom AFB, Bedford, MA, Oct. 1972.

[3] Compaq Computer Corporation, Hewlett-Packard Company, IBM Corporation et al. TCPA Main Specification Version 1.1b [EB/OL]. http://www.trustedcomputing.org/docs/main%20 v1_1b.pdf, 2002-02-22.

[4] Compaq Computer Corporation, Hewlett-Packard Company, IBM Corporation et al. TCPA PC Specific Implementation Specification Version 1.00[EB/OL]. http://www.trusted computing.org/ docs/TCPA PC-Specification_v1.00.pdf, 2001-09-09.

[5] A. Pochiraju, "Securing Networked Storage using Defense in Depth", GIAC practical repository, June 15, 2003.

[6] K. Keeton, W. John, "Automating data dependability", 10th ACM-SIGOPS European Workshop, Saint-Emilion, France, September 2002, 93-100.

[7] H. Gobioff, D. Nagle, et al, "Integrity and Performance in Network Attached Storage", Technical Report CMU-CS-98-182, December 1998.

[8] S. Kent, R. Atkinso, Security Architecture for the Internet Protocol, RFC 2401, November 1998.

[9] J. Satran, D. Smith, et al, IPS Internet Draft – iSCSI, http://www.ietf.org/internetdrafts/draft-ietf-ips-iscsi-0 8.txt, Sept, 2001.

[10] D. Mazieres, D. Shasha. "Don't trust your file server", HotOS, May 2001.

[11] E. Riedel, M. Kallahalla, et al, "A framework for evaluating storage system security", Proceedings of the 1st Conference on File and Storage Technologies (FAST), Monterey, CA, January 2002.

[12] E. Zadok, J. Nieh, "FiST: A Language for Stackable File Systems", Proceedings of the Annual USENIX Technical Conference, June 2000, pp.55-70.

Interactive System Of Image Information Based On Distribution Computing

Tianzhen Liu
College of Computer Science and Technology, Wuhan University of Technology
Wuhan, Hubei Province 430070, P.R.China
Email: bottaltz@126.com
And
Hengqing Tong, Xujie Zhao
Department of Mathematics, Wuhan University of Technology
Wuhan, Hubei Province 430070, P.R.China
Email: hqtong2005@public.whut.edu.cn

ABSTRACT

Aiming at a supposed hit-and-run affair, an interactive system of image information was proposed. Through microprocessors on CCD camera covering all roads of a city, the system can recognize vehicles by module of color recognition based on optimized support vector machine, module of logo recognition based on edge histogram and module of type recognition based on Bayesian network of incomplete data sets. The design of the system's hardware and the mechanism of software algorithm were elucidated in detail. In experiment, the system can recognize suspicious vehicles exactly according to the information provided by eyewitnesses in site.

Keywords: Distributed Computing, Pattern Recognition, Feature Extraction, and Image Information Interaction.

1. INTRODUCTION

Vehicle recognition is typical practical technology of pattern recognition technology used in field of transportation, which includes vehicle license plate recognition, vehicle type recognition, body color recognition, vehicle logo recognition and so on[1,2] . With many years' research and practical applications, the domestic technology of vehicle license recognition tends to be mature. It has been used in many kinds of intelligent transportation system [3, 4]. However, faced the situation of a changed or cloned vehicle license, the vehicle license plate recognition technology has lost its function.

In a supposed scene, a police station receives a call that a serious hit-and-run affair has happened in somewhere. The witness confirms the escaped vehicle is a red Santana but he can't assure the number of the license plate. In this instance, the vehicle's license plate number is unable to obtain, so the police can only search the suspect car according to the witness's other description. Normally it not only needs put more police into this affair, but also can hardly solve the problem in a short time. So a more generalized vehicle recognition system is needed to complete the task. This kind of system based on distributed computing thought sets center control system in the traffic management center and sets subsystems in the roads. The center control system and the subsystems exchange the image information to check the suspect cars, which can enhance efficiency greatly.

2. ANALYSIS OF TARGET CAR

License plate is an identity card of vehicle. Different vehicles have different license plates. So vehicle license plate recognition technology has been an effective method of vehicle recognition in the mature pattern recognition technologies [5]. But regarding the illegal activity related to vehicles, the car license recognition has the congenital flaw. Criminals often replace the original license plate by another one after the stealing to avoid tracing. Some men usually hang up the cloning license plate to avoid the fare and the annual inspection. In hit-and-run affairs, the witnesses are not sure to see the license number clearly. In all the situations above, license plate recognition system can't take effect.

In the conceived scene of this paper, we only know the escaping car is a red Santana. Then we design the recognition system aiming at the feature "red Santana". We can extract the features of the target car. First, the color feature is red. Second, the Vehicle Logo feature is Volkswagen. The last, the vehicle type feature is a car. In view of this, we design a pattern recognition system composed of body color, vehicle logo and vehicle type recognition, which takes advantage of full information in "red Santana".

In order to find the escaping vehicle, searching region should cover the all city. Namely, we must install a CCD camera above every crossroad of the city [6]. Moreover, the system has the rate of misidentifying and recognizes nothing sometimes, which requires interactive system of image information based on distributed computing to confirm the result repeatedly.

3. SYSTEM FRAME MODE

Vehicle flows on highways in a city everyday. To return all the vehicle images to the transportation management center and examine the suspect vehicle manually encounter bad capacity of recognition, a large amount of image information and low speed of returning [7,8]. A subsystem can be made to fix a microprocessor on each CCD camera. In this way, the center control system can transmit the colors, patterns and logo features of the suspect vehicle to the subsystem. The microprocessor of the subsystem can match the features with the vehicle pictures taken by the camera and transmit the successfully matched vehicle pictures and the spots of discovering back to the center control system of transportation management center. It will largely save the manual work of the police and shorten the case-solving time. Herein, the following system frame mode is designed:

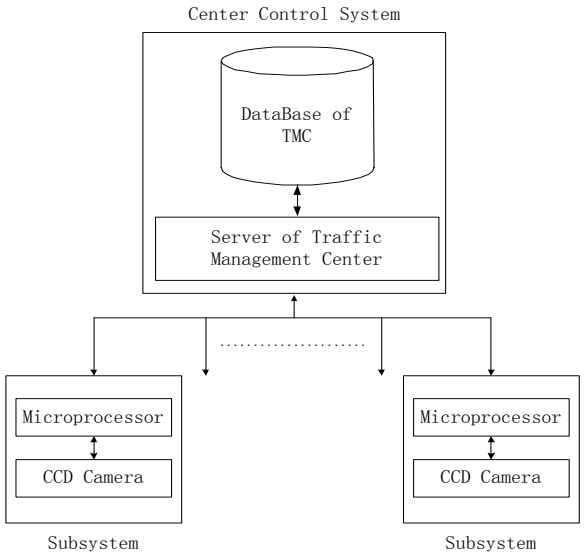


Fig. 1. System framemode

4. SOFTWARE MODULE DESIGN AND SPECITICATION

In some scenes similar with the ones in this paper, the witness may not see the vehicle number clearly. But the color, mark and type of the vehicle are easy to remember. This system, according to this circumstance, designs the software integrating three modules of color recognition, logo recognition and type recognition to minimize the range of manual examination.

4.1 Color Recognition

We take the method of optimized support vector machine (SVM) [9] for vehicle color recognition because the SVM is independent of samples' total number, which has high speed and simple structure. The module of vehicle color recognition is used in microprocessor of subsystems.

We proposed the improved SVM algorithm according to this system. The speed of classification of the multi-type SVM algorithm depends on the amount of SVM n_{SVM}. In the classification problem of 16 types determined by literature n_{SVM} = 120, the amount of SVM must be reduced to improve the classification speed [10]. We solve the problem by combining colors and decomposing the sample space.

1) Colors combining:

In color space of Lab**,** white, grey, silver and black are close in the value of *a* and *b* (color area's projected value on the *ab* plane of Lab color space) and are mainly distinguished by *L*. Therefore, we combine them into one color denoted by grey. Theoretically, the defining field of combined grey is larger than the defining fields of the four separate colors and will affect the neighbor colors of the four. The experimental result shows that this affection is not enough to produce wrong classification result because no neighboring color sample is misclassified as grey. The types of color are reduced from 16 to 13 by color combining.

2) Sample space decomposing: In the HSV color space, *H* represents the main color tone and *V* represents the purity degree of color [11]. When *V* is not small, the value of *H* plays an important role in the naming of colors. For example, when $0 \leqslant H < 60$, it is impossible to be dark blue, light blue or purple whatever the values of *S* and *V* are; in the color subclass limited by $0 \leqslant H < 60$, the types of dark blue, light blue and purple can be wiped off. Therefore we can decompose the sample space to several subspaces according to the value of *H*, each space with fewer types of colors. We chiefly decompose the sample space to six subclasses according to the value of *H*. The situation of color types and number of samples in every sub-space are shown in Fig. 1.

Table1. Table of color space decomposition

Sequence Number	Rage of Feature Value	Number of SVM	Number of Samples
1	$0 \leq H < 60$, $S \geq 50$	21	24
2	$0 \leq H < 60$, $S < 50$	21	30
3	$60 \leq H < 120$	15	34
4	$120 \leq H < 240$	12	33
5	$240 \leq H < 300$	6	22
6	$300 \leq H < 360$	15	22

Sequence Number	Types of Color
1	red, brown, orange, yellow, milk yellow, gold, bottle green
2	red, brown, milk yellow, gold, dark green, pink, grey
3	Yellow, milk yellow, gold, dark green, light green, grey
4	dark green, light green, dark blue, light blue, grey, purple
5	dark blue, light blue, grey, purple
6	red, brown, pink, dark blue, grey, purple

Commonly, the more the sub-spaces of the sample space to be decomposed, the fewer the SVM is in corresponding sub-spaces, and the more complicated and time-consuming the decomposing algorithm is. The decomposing algorithm above is the combined result of balancing SVM amount and complication degree. In fact, except the 1～2 types of sub-spaces, the other sub-spaces do not need real RGB to HSV transform[12] . They can be ascertained only by comparing the size of RGB. This saves time for the sub-class decomposition.

The vehicle color recognition module based on the optimized SVM can recognize the color of the vehicle efficiently with the average speed of 0.31 sec for each and with the ratio of 88.6% of correctness.

4.2 Vehicle Logo Recognition

It takes the method of logo recognition based on the edge histogram to collect logo for template matching [13].

First save edge histograms of many logo templates in computer as $M_t(i)$ and then calculate the edge histogram of the logo to be recognized as $h(i)$.

$$D(t) = \sum_{i=1}^{36} \left| M_t(i) - h(i) \right|, \ t = 1, \mathrm{L}\ , T \qquad (1)$$

In Eq.(1), T is the number of templates and $D(t)$ is the otherness between the edge histogram of the logos to be recognized and the edge histogram of every logo template. Well the logo is $\arg\min D(t)$.

According to the present situation of China, 24 kinds of templates are made with the correct ratio being 90% and recognizing time being 1.73s.

4.3 Vehicle Type Recognition

Nowadays vehicle recognition technology can only achieve "broad types of identification". The types include car(C), truck (T), heavy truck (H.T) and noise (N). This paper adopted the method for classifying vehicles with Bayesian network based on incomplete data sets to design a module of vehicle type recognition, which used in the microprocessor [14].

1) The modular structure of vehicle type recognition with IDS-BN:

Fig. 2. is the schematic drawing of Vehicle Type Recognition with IDS-BN. First, it collects images and decodes them with MPEG-2, then pretreats the images such as the image restoration, the image division, the two-value transforming, and the edge withdrawing [15]. The pretreatment can filter out the interference, the noise, so as to find out the clear outline of the vehicle. Second, it withdraws the valid feature of the vehicle type and set up the model of the vehicle type recognition with IDS-BN based on the characteristic parameter, so as to establish the model parameter of the study network. In the last, the model and parameters are input into the classifier of vehicle type based on IDS-BN in order to classify the vehicle type with the image of the vehicle type output.

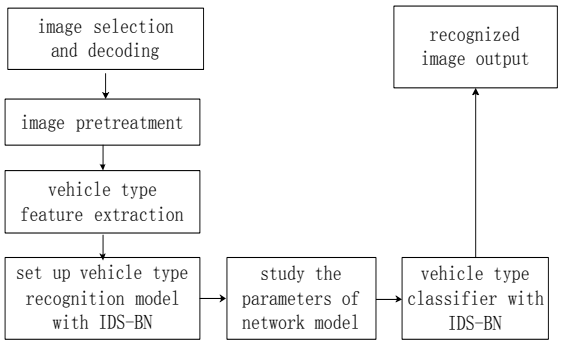


Fig. 2. Module structure of vehicle type recognition with IDS-BN

2) Feature extraction of the vehicle type based on IDS-BN:

It is impossible to obtain the necessary characteristic variable because the data set of training samples is incomplete. How to extract simple and valid vehicle type is the chief problem that the vehicle type recognition system with IDS-BN should solve. We can know from the prior data that the network model of vehicle type recognition can extract characteristic variables as follows in the Table 2..

The six characteristic variables are $Type$, x_i, y_i, v_x, v_y and l/w in the network model. The relationship of the above variables in the vehicles has relatively stable structure and appearance. This structure's inherent attribute insures the feasibility of the structure studying. Otherwise, to study network structure from the data is not only complicated, but expensive in the network's maintenance. Moreover, its estimate parameters are excessive; the variance of system is bigger, which influence the predicted precision.

Table 2. Model variables of vehicle type recognition

Name of variables	Instruction of variables
$Type$	vehicle type waiting to be recognized
x_i	x-coordinate of the vehicle's location
y_i	y-coordinate of the vehicle's location
v_x	vehicle's velocity in *x* direction
v_y	vehicle's velocity in *y* direction
l	length of the vehicle
w	width of the vehicle
l/w	ratio of length to width of the vehicle
$perimeter^2/area$	ration of perimeter's square to area of the vehicle
$area \times area$	square of area of the vehicle
$camera\ zoom$	magnified multiple of the camera

The method for classifying vehicles with Bayesian network based on incomplete data sets can recognize the broad type of the vehicles with the average ratio higher than 90% of correctness. The ratio of correctness for car recognition can reach 96.2%.

5. CONFIRM THE SUSPICIOUS VEHICLE BASED ON DISTRIBUTED COMPUTING

The system has a certain rate of misidentifying. So it is necessary for the subsystems to confirm the escaping car repeatedly. In addition, the system needs to trace it after finding the suspicious vehicle. All of these have provided condition for the application of the distributed computing thought [16]. The following is the elaboration of distributed system's movement.

Step 1: According to the description of the witness, the police input in the center system three conditions which are 'body color=red, vehicle logo=Volkswagen, vehicle type=car'.

Step 2: The center system converts the conditions into feature data. The range of color feature: $0 \leqslant H < 60$, $S \geqslant 50$; $0 \leqslant H < 60$, $S < 50$; $300 \leqslant H < 360$. The feature of vehicle logo is: the template of Volkswagen. The feature of vehicle type is: x_l (the *x*-coordinate of the Santana's location)、y_l (the *y*-coordinate of the Santana's location)、v_x (the Santana's velocity in *x* direction)、v_y (the Santana's velocity in *y* direction)、l_l (the length of the Santana)、w_l (the width of the Santana)。

Step 3: Every subsystem extracts the features of the

vehicles, which appear in its monitoring scope, then match them with the features of the escaping vehicle. If matching successfully, the subsystem will identify the license plate with the vehicles license plate recognition mode. And

The variable i, which is tied to its license numbers. The weight of i is set as $\omega_i = 1$.

Step 4: The CMS passes the serial number and license number to the next subsystem by which the suspicious vehicle will pass, and it orders the subsystem to match the vehicle again.

Step 5: If matching successfully, then turn to Step 4. The CMS add 1 to the weight of this vehicle ($\omega_i = \omega_i + 1$).

Step 6: If the matching is failed, the subsystem passes the image of the vehicle to the CMS. Then human identifies it. If it is a suspicious one, the CMS maintain its order number, turn to Step 4. If it is not a suspicious one, the CMS delete the data about this vehicle.

Step 7: The police can order the subsystem to submit the image of high weight vehicle by CMS at any time.

Step 8: The police station can concentrate police forces to search the escaping car among the limited suspicious vehicles that the system picks up.

6. CONCLUSION

The vehicle license plate recognition is not a panacea among the methods of vehicle recognition. Under the situation that the vehicle's image or the information of license plate can't be obtained, this paper proposed an interactive system of image information based on distributed computing thought to complete the task efficiently instead of vehicle license plate recognition. Especially in the hit-and-run affairs, the system not only saves the police forces but also solves the case in a short time.

7. REFERENCES

[1] Sergios Theodoridis, Konstantinos Koutroumbas, Pattern Recognition, Second Edition, and United Kingdom, Elsevier Science, 2003.

[2] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford: Clarendon Press, 1995.

[3] D.D.Zhang, P.Fu, et al, "Traffic peccancy processing system based on image recognition", The Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, 2003, 1200-1205.

[4] D. Koller, "Moving object recognition and number of license plate and the traveling direction will be submitted to the center management system. The CMS numbers the suspicious vehicle by its submitted order with

classification based on recursive shape parameter estimation", Isreal, Proc 12th Conf. Artificial Intelligence, Computer Vision, Dec. 1993, 27-28.

[5] P. Kumar, S. Ranganath, et al, "Cooperative multi-target tracking and classification", In Proc. Eur. Conf. Computer Vision, 2004 (5): 376-389.

[6] R. Collins, A. Lipton et al, A System for Video Surveillance and Monitoring: VSAM Final Report[R]. Carnegie Mellon University, Technical Report: CMU-RI-TR-00012, 2000.

[7] L.V. Dutra, R. Huber, "Feature extraction and selection for ERS-1/2 InSAR classification" [J], International Journal of Remote Sensing, 1999, 20(5): 993-1016.

[8] I. Haritaoglu, D. Harwood, et al, "Real-time Surveillance of People and Their Activities" [J], IEEE Trans Pattern Analysis and Machine Intelligence, 2000, 22 (8): 809-830.

[9] Y.Q.Wang, Z.S.You, et al, "Vehicle Color Recognition Using Support Vector Machine" [J], Journal of Computer-Aided Design and Computer Graphics, 2004, 701-706.

[10] The register standard of motor vehicles [S]. Beijing: Traffic Management Bureau of The Ministry of Pubic Security of China, 2001.

[11] B Chakraborty, Genetic Algorithm with Fuzzy Fitness Function for Feature Selection, Proc of the 2002 IEEE Int'l Syrmp on Industrial Electronics, 2002, Vol.1: 315-319.

[12] S.B. Serpico, L. Bruzzone, "A New Search Algorithm for Feature Selection in Hyper Spectral Remote Sensing Images", IEEE Trans on Geo-science and Remote Sensing, 2001, 39(7): 1360-1367.

[13] B.Luo, Z.S.You et al, "The Fast Method in VLR Based on Edge Directions" [J], Application Research of Computers, 2004, 150-157.

[14] G.L.Yun, Q.M.Chen et al, "Method for Classifying Vehicles with Bayesian Network Based on Incomplete Data Sets" [J], Computer and Communications, 2006, 66-69.

[15] A.L. Blum, P. Langley, "Selection of Relevant Feature and Examples in Machine Learning", Artificial Intelligence, 1997, 245-271.

[16] B. Rajkumar, High Performance Cluster Computing: Programming and Applications, New York: Prentice-Hall Inc., 2001.

Implementation of a Distributed Coupled Atmosphere-Wave-Ocean Model: Typhoon Wave Simulation in the China Seas*

Jinfeng Zhang1,2
1 Navigation College, Wuhan University of Technology
Wuhan, Hubei, China
2LASG, Institute of Atmospheric Physics, Chinese Academy of Sciences
Beijing, China
Email: mount@mail.whut.edu.cn
Liwen Huang[1,2,3]
1Wuhan Heavy Rain Institute
Wuhan，Hubei, China
2Navigation College, Wuhan University of Technology
Wuhan, Hubei, China
3LASG, Institute of Atmospheric Physics, Chinese Academy of Sciences
Beijing, China
Email: lwhuang@mail.whut.edu.cn
Yuanqiao Wen
Navigation College, Wuhan University of Technology
Wuhan, Hubei, China
Email: wenyuanqiao@yahoo.com.cn
Jian Deng
Navigation College, Wuhan University of Technology
Wuhan, Hubei, China
Email: djcharm@mail.whut.edu.cn

ABSTRACT

A distributed coupled framework system was developed. The framework consists of atmosphere model (MM5), ocean model (POM), wave model (WAVEWATCH), and a coupled program (coupled module). The coupled program controls Atmosphere-Wave-Ocean model calculations and data exchanges. The impact of air-sea-wave interaction was considered to examine the validity and performance of this coupled model. A typical typhoon case Vongfong (No0214) was modeled to examine the wave simulation validity in the China Seas. The coupled model can obviously improve the precision of the significant wave height (SWH) as compared to the uncoupled run.

Keywords: Distributed Framework; Coupled Atmosphere-Wave-Ocean Model; Typhoon; Wave Simulation; the China Seas.

1. INTRODUCTION

The air-sea interaction of the atmospheric and the oceanic circulation is strongly influenced by the presence of property fluxes at the air-sea interface, such as the momentum flux and the heat flux. Therefore, coupled of atmosphere, wave and ocean models will increase the accuracy of the predicted physical variables, which has been found to simulate the circulation much more faithfully [1].

Different technical methods to couple the models can be selected depending on the models used and technical resources. Generally, coupled model is performed in one of the three following ways. The traditional way is through file I/O as shown by Blain [2] and Hoder [3]. In this case the models are left relatively unaltered and are executed for a very short length of time. Model preprocessors then transform the output files of the first model into the input files of the second model. Depending on the frequency of coupling, this can be a very costly alternative. However, for very slowly varying physics this is a suitable methodology [4].

The second method of model coupling, subroutinazation, requires one of the models to be written as a subroutine of the other model [4]. This will allow for simple exchange of information between models through argument lists. This requires the two models to run on the same hardware platform and under the same parallelization paradigm which can significantly reduce model performance. Furthermore, this approach requires significant modifications to the two models. Therefore, it produces a code which is difficult to maintain.

The final common method for model coupling is through an MPI interface [5]. In this method, calls are added to both applications to send data to each other or in an abstract form with the Model Coupled Toolkit. This has several advantages over the previous two methods: First, because the models are only started once, this reduces the overhead when compared to the file coupled approach. Second, this approach requires much less modification of the given applications allowing for better maintainability of the models. Third, with distributed MPI implementations, it is feasible to distribute applications on different hardware platforms for improved model performance [4]. However, this approach does suffer from two problems: First, this method does not allow for permanent storage of coupled information, thus all models must be executed simultaneously, which is not required for one-way coupling. Second, since MPI implementations spin lock while waiting for data, it is not advisable to over-subscribe CPUs on a

*This study is jointly supported by National Natural Science Foundation of China (NO:40275015) and Shanghai Typhoon Research Foundation

single system when working on a leap-frog algorithm [4].

In Section 2 the models and the agent-based distributed coupled framework are described. A typical typhoon case will be chosen to examine the validity of this coupled model in Section 3. In Section 4 the results of the coupled model are given, and then compared with the TOPEX/Poseidon altimeter data. In Section 5 the conclusions are summarized and future plans are presented.

2. DISTRIBUTED COUPLED FRAMEWORK

In this coupled framework, calculations of three models (atmosphere model, wave model and ocean model) are carried out as independent tasks for different processors and a coupled central server controls the data exchanges among models using different agents respectively [6]. The whole system is a multi-agent system (MAS) [7], All these mutually independent agents (model agents, coupler agents, system management agents and resources management agents, and etc) construct a virtual coupled computing space, which shields the heterogeneity of the system and the complexity of the application environments and makes all users share a single and transparent working space for their own domain.

The coupled operation can be implemented as follows: At first, the coupled central server is started, and then the models are performed one by one. The central server controls data exchanges and calculation processes among three models, it receives 2D or 3D-field data from a certain model and distributes them to other models in arbitrary time intervals, which are prescribed in the certain models. If model grids are different between sender and receiver models, the 2D or 3D-field data should be interpolated from the grid of a sender model to the grid of a receiver model. Therefore, this coupling has flexibility to use different resolution of grid and time step for each model. The source code of each model can be modified simply, added some data exchange routines in the original model which call the coupled routines. Each model code keeps its original structure.

Fig. 1. shows the interactions among three numerical models: atmosphere model (MM5v3), wave model (WAVEWATCH-III 2.22) and ocean model (POM2k), each model exchanges the data with the central server respectively and mutually. The MM5 provides the 10m surface wind field to generate wind-wave in WAVEWATCH, and offers surface momentum fluxes and heat fluxes to the POM; The surface roughness length and wave-induced stress, which are calculated by the WAVEWATCH, are offered to MM5 and POM; For the POM, the calculated surface current fields and water elevation are provided to the WAVEWATCH, and the sea surface temperature (SST) from POM is used by MM5 [8].

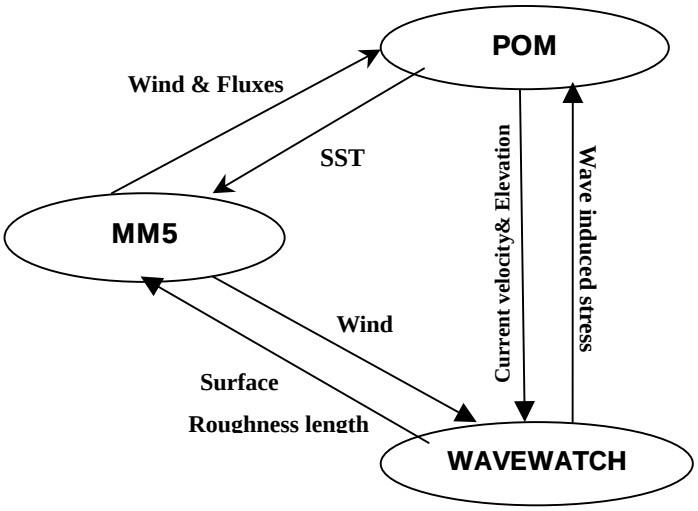


Fig. 1. Air-sea-wave interactions in the coupled system

3. TYPHOON CASE

A typical typhoon case Vongfong (No0214) was simulated to examine the validity and importance of this coupled model. Tropical Cyclone Vongfong strengthened to tropical storm at 0200 UTC 18 August 2002. Vongfong moved on a general northwest heading reaching maximum strength of 30 m/s and a central pressure of 980 hPa, then Vongfong made landfall near Wuchuan, Guangdong Province, at approximately 2000 UTC 19 August. Finally, Vongfong weakened into a tropical depression in Guangxi Province.

The experiment was run on a grid of 90x110 points with a horizontal resolution of 37.5km for MM5, and the time step was selected as 120 seconds. A horizontal grid dimension of 193x241 with a resolution of 1/6°x1/6° was used for both POM and WAVEWATCH, and time step was selected as 480 seconds. The atmosphere model was executed on 8 CPUs of an SGI Origin 300, the ocean model was run on one PC, while the wave model was run on a Linux workstation and the coupled server was executed on the same Linux system. The models were coupled every 480 seconds.

4. WAVE SIMULATION RESULTS

The maximum significant wave height (SWH) is found at August 19th, 00UTC, the simulated SWH for the uncoupled and coupled simulations of Tropical Cyclone Vongfong is shown in Fig. 2., the SWH simulated by the WAVEWATCH model individually is given in Fig. 2.a, and Fig. 2.b shows the MM5-POM-WAVEWATCH coupled results. As compared with the uncoupled results in Fig. 2.a, the SWH maximum is considerably improved in the coupled simulation (Fig. 2.b), due to the consideration of the air-sea-wave interactions.

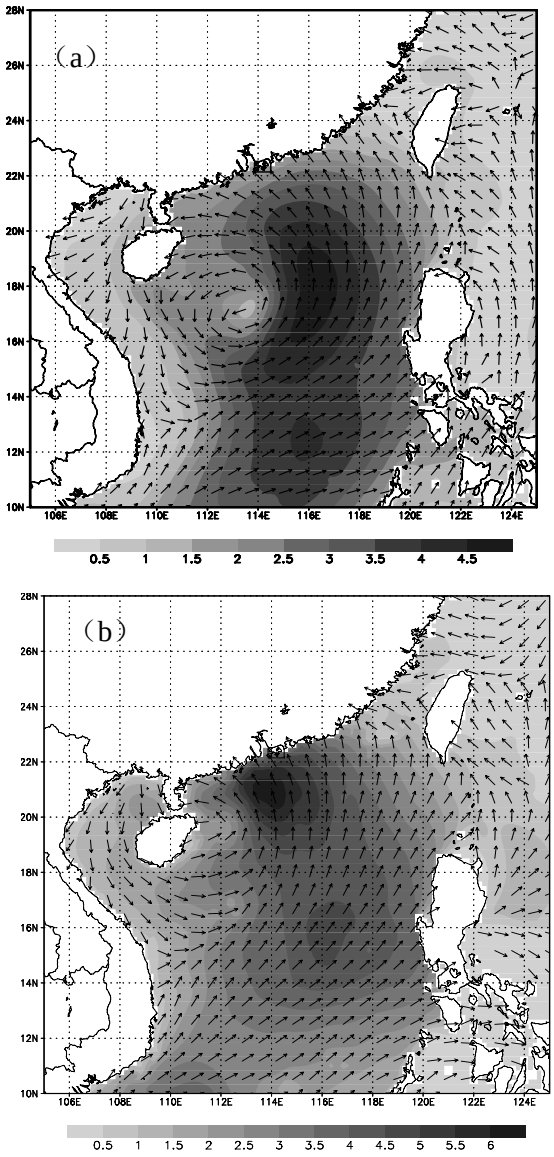


Fig. 2. Simulated significant wave heights for 24 hours at August 19th, 00UTC by uncoupled wave model (a) and coupled model (b)

The simulated SWH is compared with the SWH observed by TOPEX/Poseidon altimeter (Fig. 3.), there is a good agreement between the observed and the modeled data. Compared to the uncoupled case, the coupled simulation is much closer to the TOPEX/Poseidon wave field.

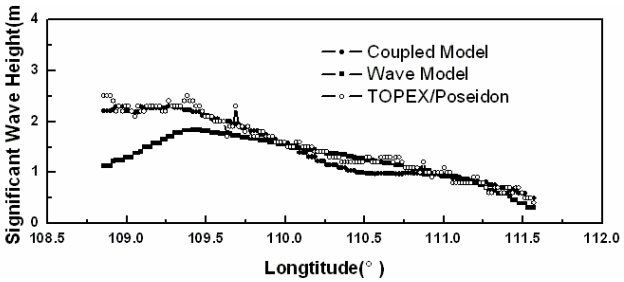


Fig. 3. Comparison of SWH simulated by the coupled model, wave model and the observed TOPEX/Poseidon data

5. CONCLUSIONS

A distributed coupled framework system was developed and test calculation was carried out for the wave field simulation around the China Seas. The framework consists of models for the atmosphere (MM5), ocean (POM), and wave (WAVEWATCH), and a coupled central server that controls calculations of these models and data exchanges, and it utilizes agent-based infrastructure to aid in the process of coupled.

The impact of air-sea-wave interaction was considered to examine the validity and performance of this coupled model system, using the mesoscale short-term forecasts of the typhoon Vongfong in the China Seas. According to the comparison between the modeled results and the observed results, the coupled MM5-POM-WAVEWATCH simulation improves the precision of the SWH. The air-sea-wave interaction involved is very useful to improve the forecast precision. For the physical scheme of the coupled model, more typhoon cases should be chosen to verify the performance. For the computational aspect, improvements are necessary for distributed scheme and data exchanges among MM5, WAVEWATCH, and POM to achieve better computational load balance.

6. REFERENCES

[1] W. Zhao, S. Chen, "A Coupled Atmosphere-Wave-Ocean Framework for High-Resolution Modeling of Tropical Cyclones and Coastal Storms," Rosenstiel School of Marine and Atmospheric Science, University of Miami.

[2] C. Blain, M. Cobb, "Wave-current interaction in a wave-breaking environment," Recent Advance in Marine Science and Technology, 2001.

[3] R. M. Hodur, "The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System COAMPS," Monthly Weather Review, 1996.

[4] M. T. Bettencourt, "Distributed Model Coupled Framework," Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11 2002 (HPDC'02), 2002.

[5] H. Nagai, T. Kobayashi, et al, "Coupled Atmosphere, Land-surface, Hydrology, Ocean-wave, and Ocean-current Models for Mesoscale Water and Energy Circulations," Japan Atomic Energy Research Institute.

[6] L.W. Huang, J. Deng, et al, "Coupled Air-Sea-Wave Mesoscale Model System and its Experiments of Typhoon over the South China Seas," The Sixth National Conference on Cyclone Meteorology, Chengdu, China, August 2005.

[7] Y.Q. Wen, S.S. Yu, et al, "Agent Based Distributed Parallel Tunneling Algorithms," Lecture Notes in Computer Science, 2004, 226-230.

[8] L. N. LY, "A Numerical Algorithm for Solving a Coupled Problem of the Air-Sea-Wave Interaction," Mathl. Comput. Modelling, 1996, 24(7): 19-32.

Distributed Computing In Multi-Group Structural Equation Model

Hengqing Tong, Ziqing Li
Department of Mathematics, Wuhan University of Technology
Wuhan, Hubei Province 430070, P.R.China
Email: hqtong2005@public.whut.edu.cn
QiaoLing Tong
Department of Electronic Science and Technology, Huazhong University of Science and Technology
Wuhan, Hubei Province 430074, P.R.China
Email: qltong@whicc.com

ABSTRACT

This paper firstly proposes an improved partial least square (PLS) algorithm in structural equation model (SEM) using a suitable iterative initial value with constraint of unit vector. The algorithm enhances the convergence rate greatly and its convergency is illuminated. Then multi-group SEM is investigated and distributed computing is adopted to calculate all the coefficients in SEM of each group, which results in considerable timesavings when compared with traditional approach. Furthermore, a uniform model is built using the generalized linear model with convex constraint and an algorithm for the multi-group SEM is presented. The results of this paper have been received in software DASC.

Keywords: SEM, PLS, Multi-Group Model, Convergency, Distributed Computing.

1. INTRODUCTION

Structural equation model (SEM) has been a booming branch of applied statistics field. It is widely applied in psychology and sociology as well as other fields, especially in Customer Satisfaction Index (CSI) model [1], which was required by a series of ISO9000 criterions. The calculation plays an important part in the applications of SEM. Many international well-known software companies have exploited software for SEM, such as LISREL, AOMS, EQS, SEPATH, MPLUS and CALIS module in SAS. Many papers focus on the improving of SEM algorithm, including EM algorithm and ML algorithm, but partial least square (PLS) is the most important algorithm in practice, in spite of its convergence of iteration cannot be ensured, or its convergence rate may be very slow [2]. Because the iterative initial value of PLS has been arbitrary in papers or in software so far, it is necessary to improve it. Many papers focus on the extension of the SEM, including multi-level SEM [3] and nonlinear SEM [4], but multi-group SEM has not been discussed so far. Having proposed a kind of generalized linear model with a convex constraint [5], we may discuss multi-group SEM in this paper.

Distributed computing is a kind of new computing aiming especially at complex scientific computation and grows rapidly with the development of Internet. In modern society, there are various subjects in each question for discussion, which covers a wide range of subjects and is classified with circumstances. Each subject seems to need substantial computation. Astronomy research organizations utilize computer to analyze space impulse and the movements of the stars; biologists use it to simulate human proteome folding; economists make use of it to analyze the direction of some enterprise under consideration of millions of factors. This shows that the future science cannot be separated from computation hourly. While because of unique advantage----cheapness and high efficiency, distributed computing has drawn more and more attention.

This paper presents an improved PLS algorithm in SEM using a suitable iterative initial value with constraint of unit vector. The algorithm enhances the convergence rate greatly and its convergency is illuminated. Then we investigate multi-group SEM and adopt distributed computing and the generalized linear model with convex constraint (Tong, 1993) to build a uniform model. The results of this paper have been received in software DASC [6].

2. STRUCTURAL EQUATION MODEL

Structural Equation Model and Partial Least Square Algorithm

There are two systems of equations in a SEM. One is a structural system of equations among structural variables, and the other is an observation system of equations between structural variables and observed variables. The Chinese Customer Satisfaction Index (CCSI) model is a typical SEM, and the corresponding equation that relates the latent variables in the model can be written as structural equations:

$$\begin{pmatrix}\eta_1\\ \eta_2\\ \eta_3\\ \eta_4\\ \eta_5\end{pmatrix}=\begin{pmatrix}0&0&0&0&0\\ \beta_{21}&0&0&0&0\\ \beta_{31}&\beta_{32}&0&0&0\\ \beta_{41}&\beta_{42}&\beta_{43}&0&0\\ 0&0&0&\beta_{54}&0\end{pmatrix}\begin{pmatrix}\eta_1\\ \eta_2\\ \eta_3\\ \eta_4\\ \eta_5\end{pmatrix}+\begin{pmatrix}\gamma_{11}\\ \gamma_{12}\\ \gamma_{13}\\ \gamma_{14}\\ 0\end{pmatrix}\xi_1+\begin{pmatrix}\varepsilon_{\eta 1}\\ \varepsilon_{\eta 2}\\ \varepsilon_{\eta 3}\\ \varepsilon_{\eta 4}\\ \varepsilon_{\eta 5}\end{pmatrix} \quad (1)$$

where $\eta_1 \sim \eta_5$, ξ_1 are structural variables, β_{ij} is the path coefficients from dependent variable η_j to η_i, and γ_{ij} is the path coefficients from independent variable ξ_j to dependent variable η_i.

In general, suppose that $\eta_1 \sim \eta_m$ are m dependent variables, arranging them as a vector η by column as

* This work is partially supported by the Science Foundation Grant No.02C262 14200218 for Technology Creative Research from the Ministry of Science and Tec_ hnology of China, the Chinese NSF Grant No.30570611 to Tong H. Q. granted by Wuhan University of Technolo_ gy in China.

Eq.(1). Analogously, $\xi_1 \sim \xi_k$ are k independent variables, arranging them as a vector ξ by column too. Thus the $m \times m$ square matrix B is the coefficient matrix of η, the $m \times k$ matrix Γ is the coefficient matrix of ξ, and ε_η is the residual vector. Therefore, Eq.(1) may be extended as:

$$\eta = B\eta + \Gamma\xi + \varepsilon_\eta \tag{2}$$

The structural variables are implicit and cannot be observed directly, and each structural variable is corresponding too many observed variables. Suppose that there are M observed variables and each one has N observed values, then we will get a $N \times M$ matrix.

The relationships between the structural variables and the observed variables can also be expressed in equations by two ways of path causality. Let $x_{tj}\,(j = 1,\mathrm{L}\ ,S(t))$ be the observed variables corresponding to $\xi_t\,(t = 1,\mathrm{L}\ ,k)$, and $y_{ij}\,(j = 1,\mathrm{L}\ ,L(i))$ be the observed variables corresponding to $\eta_i\,(i = 1,\mathrm{L}\ ,m)$, then the observation systems of equations from observed variables to structural variables are

$$\xi_t = \sum_{j=1}^{S(t)} \psi_{tj} x_{tj} + \varepsilon_{\xi t},\quad t = 1,\mathrm{L}\ ,k \tag{3}$$

$$\eta_i = \sum_{j=1}^{L(i)} \omega_{ij} y_{ij} + \varepsilon_{\eta i},\quad i = 1,\mathrm{L}\ ,m \tag{4}$$

Contrarily, the observation systems of equations from structural variables to observed variables are

$$\begin{pmatrix} x_{t1} \\ \mathrm{M} \\ x_{tS(t)} \end{pmatrix} = \begin{pmatrix} \upsilon_{t1} \\ \mathrm{M} \\ \upsilon_{tS(t)} \end{pmatrix} \xi_t + \begin{pmatrix} \varepsilon_{xt1} \\ \mathrm{M} \\ \varepsilon_{xtS(t)} \end{pmatrix},\quad t = 1,\mathrm{L}\ ,k \tag{5}$$

$$\begin{pmatrix} y_{i1} \\ \mathrm{M} \\ y_{iL(i)} \end{pmatrix} = \begin{pmatrix} \lambda_{i1} \\ \mathrm{M} \\ \lambda_{iL(i)} \end{pmatrix} \eta_i + \begin{pmatrix} \varepsilon_{yi1} \\ \mathrm{M} \\ \varepsilon_{yiL(i)} \end{pmatrix},\quad i = 1,\mathrm{L}\ ,m \tag{6}$$

where υ_{tj} and λ_{ij} are load items.

We call Eq. (2), Eq. (3), Eq. (4) (or Eq. (2), Eq. (5), Eq. (6)) a SEM, and sometimes we call it a path analysis model.

At present, the path causality in popular PLS algorithm for SEM is from observed variables to structural variables as Eq.(3)Eq.(4), and the iterative initial value for ψ_{tj} and ω_{ij} is arbitrary such as $\mathbf{1}_M = (1,0,0,\mathrm{L}\ ,0)$, where $\mathbf{1}_M$ is a unit vector with M dimensions. The iterative process can be described as follows:

$$\begin{aligned} &(\psi_{tj},\omega_{ij})^{(0)} \xrightarrow{(3)(4)} (\hat{\xi}_t^{(\text{exogenous})},\hat{\eta}_i^{(\text{exogenous})})^{(0)} \\ &\xrightarrow{(2)} (\gamma_{tj},\beta_{ij})^{(0)} \\ &\xrightarrow{(2)} (\hat{\xi}_t^{(\text{endogenous})},\hat{\eta}_i^{(\text{endogenous})})^{(0)} \\ &\xrightarrow{(3)(4)} (\psi_{tj},\omega_{ij})^{(1)} \end{aligned} \tag{7}$$

where endogenous means the parameter estimation is obtained from structural equation, and exogenous means the parameter estimation is obtained from observed equation. As mentioned above, the convergence of the PLS has not been proved well, and its convergence rate may be very slow.

The LS Solution of SEM with Constraint of Unit Vector

We find that arbitrary initial value is not necessary and we can calculate PLS by a suitable iterative initial value based on least square estimation in the observation equations. First of all, let us specify some essential properties of SEM, which are embodied in three aspects as follows:

Firstly, the solution of model consisting of Eq. (2), Eq. (3), and Eq. (4) is not unique. If η_t, ξ_i are the solutions of the model, $c\eta_t, c\xi_i$ are solutions of the model too, where c is a constant. So we can solve the model in the constraint condition of unit vector.

Secondly, there exists zero solutions in Eq.(2), Eq.(3) and Eq.(4), but neither in Eq.(5) nor Eq.(6).

Thirdly, the system of Eq. (4) is equivalent to the system of Eq. (6) on condition that $\sum_{j=1}^{L(i)} \omega_{ij}\lambda_{ij} = 1$. So the solution of Eq. (4) is also the solution of Eq. (6). In the past, the PLS algorithm has been based on Eq. (3) Eq. (4) but not Eq. (5), Eq. (6). However, in this paper we deduce the iterative initial value based on Eq. (5) and Eq. (6).

Let the left vector in Eq. (6) be y_i, similarly we get η_i and ε_i, then Eq. (6) can be written as $y_i = \lambda_i\eta_i + \varepsilon_i, i = 1,\mathrm{L}\ ,m$. Multiply y_i' by y_i, we get $y_i y_i' \approx \lambda_i\eta_i\eta_i{}'\lambda_i' = \eta_i\eta_i'\lambda_i\lambda_i'$. If the structural variable is a unit vector, i.e. $\eta_i\eta_i' = 1$, then we get $y_i y_i' \approx \lambda_i\lambda_i'$. This is an approximate equivalent between two $L(i) \times L(i)$ matrices，which can be written in detail:

$$\begin{aligned} &\begin{pmatrix} y_{i1}y_{i1}' & y_{i1}y_{i2}' & \mathrm{L} & y_{i1}y_{iL(i)}' \\ y_{i2}y_{i1}' & y_{i2}y_{i2}' & \mathrm{L} & y_{i2}y_{iL(i)}' \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ y_{iL(i)}y_{i1}' & y_{iL(i)}y_{i2}' & \mathrm{L} & y_{iL(i)}y_{iL(i)}' \end{pmatrix} \\ &\approx \begin{pmatrix} \lambda_{i1}^2 & \lambda_{i1}\lambda_{i2} & \mathrm{L} & \lambda_{i1}\lambda_{iL(i)} \\ \lambda_{i2}\lambda_{i1} & \lambda_{i2}^2 & \mathrm{L} & \lambda_{i2}\lambda_{iL(i)} \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ \lambda_{iL(i)}\lambda_{i1} & \lambda_{iL(i)}\lambda_{i2} & \mathrm{L} & \lambda_{iL(i)}^2 \end{pmatrix} \end{aligned} \tag{8}$$

Note that the left elements are products of two vectors, while the right ones are products of two numbers. We take diagonal elements to be equal and get the estimation

$$\lambda_{ij}^2 = y_{ij}y_{ij}',\quad j = 1,\mathrm{L}\ ,L(i),\ \ i = 1,\mathrm{L}\ ,m \tag{9}$$

So we get the estimation of $\hat{\lambda}_{i1},\mathrm{L}\ ,\hat{\lambda}_{iL(i)}\,(i = 1,\mathrm{L}\ ,m)$.

Then we estimate the structural variable η_i. Let $\eta_i = (\eta_{i1},\eta_{i2},\mathrm{L}\ ,\eta_{iN})'$, we estimate its components one by one. For the $s'th$ component of η_i,

$$y_{ijs} \approx \lambda_{ij}\eta_{is},\quad j = 1,\mathrm{L}\ ,L(i),\quad s = 1,\mathrm{L}\ ,N \tag{10}$$

Its matrix form is

$$\begin{pmatrix} y_{i1s} \\ \mathrm{M} \\ y_{iL(i)s} \end{pmatrix} \approx \begin{pmatrix} \lambda_{i1} \\ \mathrm{M} \\ \lambda_{iL(i)} \end{pmatrix} \eta_{is}, \quad s = 1,\mathrm{L}\ ,N \tag{11}$$

Denote the vectors $Y_s = (y_{i1s},\mathrm{L}\ ,y_{iL(i)s})'$, which is the cross vector of the matrix y_i. According to OLS principle, we can obtain the estimate of η_{is}

$$\lambda_i'\lambda_i\eta_{is} \approx (\lambda_{i1},\mathrm{L}\ ,\lambda_{is})\begin{pmatrix} y_{i1s} \\ \mathrm{M} \\ y_{iL(i)s} \end{pmatrix} = \lambda_i' Y_S \tag{12}$$

$$\hat{\eta}_{is} = \frac{\hat{\lambda}_i' Y_S}{\hat{\lambda}_i'\hat{\lambda}_i}, \quad s = 1,\mathrm{L}\ ,N \tag{13}$$

where $\hat{\lambda}_i$ has been estimated above. In this way we obtain the estimation of all structural variables. They satisfy

$$\| \eta_i - \sum_{j=1}^{L(i)} \omega_{ij} y_{ij} \| \to \min$$

The geometric meaning of Eq.(13) is to find the distance between a unit spherical surface and a linear subspace. Its solution is unique if there are not linear correlations among y_{ij}.

It is necessary that there exist no common points between the unit spherical surface and the linear subspace. Otherwise, the left of Eq.(13) may be zero, namely Eq.(4) has a precise solution or satisfies $\eta_i = \sum_{j=1}^{L(i)} \omega_{ij} y_{ij}$, this means that there are linear correlations among y_{ij}.

Return to SEM, we can get the unique solution of path coefficient B and Γ. The transpose of Eq.(2) is

$$(I - B)\eta = \Gamma\xi + \varepsilon_\eta \tag{15}$$

For CCSI model, matrix B is always a lower triangular matrix whose elements in main diagonal line is zero in structural equations. So $B_I = I - B$ is also a lower triangular matrix whose determinant is 1. Therefore it is reversible, then

$$\eta = B_I^{-1}\Gamma\xi + \varepsilon_B = B_\Gamma\xi + \varepsilon_B \tag{16}$$

where $B_\Gamma = B_I^{-1}\Gamma$. So for given η and ξ, the solution based on LS in structural equations exists. If column vectors of matrix Γ are linearly independent, then the solution based on LS is still unique.

Above iterative process may be expressed as

$$(x_{tj}, y_{ij}) \xrightarrow{(5)(6),\|\xi\|=1,\|\eta\|=1} (\hat{\xi}_t^{(\text{exogenous})}, \hat{\eta}_i^{(\text{exogenous})})$$

$$(\hat{\xi}_t^{(\text{exogenous})}, \hat{\eta}_i^{(\text{exogenous})})^{(0)} \xrightarrow{(2)} (\gamma_{tj}, \beta_{ij})^{(0)}$$
$$\xrightarrow{(2)} (\hat{\xi}_t^{(\text{endogenous})}, \hat{\eta}_i^{(\text{endogenous})})^{(0)} \xrightarrow{(3)(4)}$$
$$(\psi_{tj}, \omega_{ij})^{(1)} \xrightarrow{(3)(4)} (\hat{\xi}_t^{(\text{exogenous})}, \hat{\eta}_i^{(\text{exogenous})})^{(1)} \tag{17}$$

The simultaneous equation model is transformed into an ordinary regression analysis model in Eq. (16), and it is beneficial for us to analyze and to illuminate the convergence of PLS algorithm. However, we cannot adopt Eq. (16) in practical calculation, because some elements in matrix B and Γ must be zero in Eq. (1) or Eq. (2). We still adopt Two-stage LS of the general simultaneous equation model to improve the convergence of estimation, just as we have done in DASC software compiled by us.

3. MULTI-GROUP STRUCTURAL EQUATION MODEL AND DISTRIBUTED COMPUTING

We consider more practical structure equation models now. In practice, the index measuring including the CSI measuring sometimes is carried on to several groups in the meantime. We should build up a unified measuring model, whose coefficients should be consistent for all groups, reasonable and objective.

Suppose there are W groups to be measured. Each group satisfies SEM established in Section 1 and connects with M observed variables. For each observed variables there are N observations. For each group we get a $N \times M$ data block. We accumulate these data block one by one and get a $WN \times M$ matrix together. Perhaps the model seems not complicated, but it is difficult to calculate. If the data module is dealt with respectively, the coefficients of the models are not consistent with each other and it will take long to finish the computation. How do we build a uniform model within a shorter period of time?

Distributed computing is a process through which a set of computers connected by a network is used collectively to solve a single problem. It's a hot technology applied to the field of super computer and has been used by a lot of famous system of super computer. A great deal of data is divided into small data blocks and sent to each client by server. The data is analyzed and the results are automatically sent to server. Then server unifies all the results and draws a conclusion. The parallel implementation takes better advantage of the computational power of a distributed computer system and leads to considerable speedup.

Any computer in the network can be taken as a server, which carries out the computation of main process, while the other performs, distributed computing under its direction and supervision. It is well known that the more computers the distributed computing network contains, the stronger the computing power of the network will be. Because of its flexibility in deployment and space, distributed computing is generally welcomed by the high-end users, there are considerable calculation projects such as search for Mason prime number, aliens, AIDS drugs, future weather simulation and so on, which adopt distributed computing with open structure simply. One can download procedure to perform the computation as long as he or she has a computer to connect to the Internet.

We will build a uniform model using distributed computing and the generalized linear model with convex constraint [5]. There are $W+1$ computers in the computer system and one of them will be regarded as a server and the rest as clients. Considering simpleness and reliability of the system, we make an assumption that there is intercommunication between the client and server so that they can exchange messages with each other, while there is no way between the clients. Executable files (computing task), data files (to calculate the radix), control information and detection signals are sent from the server to client, while the computing results and status messages will be sent from the client to server again. Computing data of W groups stored in the server will be sent to each client involved. The

clients will apply the structural equation model in Section 1 and its algorithm in Section 2 to calculate the coefficients of latent variables, regression coefficients between observation variables and structural variables, impact coefficients and scores of each observation variables to customer satisfaction of the corresponding group respectively. Then all the clients send the results to the server respectively. Then we get W groups of coefficients and W evaluation scores, which are inconsistent. Now we will adopt the generalized linear model with convex constraint to build a uniform model.

4. THE LS SOLUTION UNDER CONVEX CONSTRAINT IN SEM FOR MULTI-GROUP EVALUATION

For solving SEM with multi-group evaluation, we need to understand the generalized linear model with convex constraint for multi-group evaluation, which is our work earlier [5]. We relate the construction of the model and the concerned main theorems.

Let p indexes be variables, denoting them as $x_{(1)},\mathrm{L}\ x_{(p)}$. An evaluation table is an observation for a group. If we observe $i=1,\mathrm{L}\ ,N$ times for $k=1,\mathrm{L}\ ,W$ groups and for $j=1,\mathrm{L}\ ,p$ indexes, we can get the data $x_{ij}^{(k)}$, which are the elements of the matrix X. In the matrix X, the observation for a time is a row, the observation for an index is a column, and the observation for a group is a block. The weight coefficients $\beta_1,\mathrm{L}\ \beta_p$ are undermined, but they need to satisfy $\beta_j \ge 0, j=1,\mathrm{L}\ ,p$ (i.e. $\beta \ge 0$) and $\beta_1+\beta_2+\mathrm{L}\ +\beta_p=c$ (i.e. $\mathbf{1}_p'\beta=c$). This is a convex constraint. For each group we must give an evaluation score $y_i(i=1,\mathrm{L}\ ,W)$, which are also unknown and undermined beforehand. We build the model as follows and call it generalized linear model with convex constraint:

$$GP\begin{cases}\| Dy-X\beta\|^2 \xrightarrow{y,\beta} \min \\ \mathbf{1}_p'\beta=c \\ \beta\ge 0\end{cases} \tag{21}$$

where $Dy=y\otimes\mathbf{1}_p$, $D_{Wp\times W}=I_W\otimes\mathbf{1}_p$, $\mathbf{1}_p=(1,\mathrm{L}\ ,1)'$, $y=(y_1,\mathrm{L}\ ,y_W)'$. We take the average values for each data block by columns and get the compressed data matrix $X^{(0)}=\frac{1}{N}(I_W\otimes\mathbf{1}_p)X=\frac{1}{N}D'X$. Now we relate the first theorem:

Theorem 1. Let $P_D=I_{WN}-\frac{1}{N}DD'$, if $rk(P_DX)=p$, then under the constraint $\mathbf{1}_p'\beta=c$, the quadratic $Q(y,\beta)=\| Dy-X\beta\|^2 \xrightarrow{y,\beta}\min$ has unique solution: $\hat{y}=X^{(0)}\hat{\beta}$, here $\hat{\beta}=\frac{A^{-1}\mathbf{1}_p}{\mathbf{1}_p'A^{-1}\mathbf{1}_p}$, $A=X'X-NX^{(0)'}X^{(0)}=X'P_DX$. If each component of $\hat{\beta}$ is nonnegative, then $\hat{\beta}$ is the solution of GP.

If some component of $\hat{\beta}$ in the Theorem 1 is negative, about the existence and uniqueness of the solution of GP model we have the second theorem:

Theorem 2. If $rk(DMX)=W+p$, then there exists $\hat{\beta}_0$ which is a unique solution of GP model. If some component of $\hat{\beta}$ in the Theorem 1 is negative, then $\hat{\beta}_0$ have 0 components, whose location is the same as the location of the negative component of $\hat{\beta}$.

Theorem 2 gives the existence and uniqueness of the solution of GP model, but we have not had a concrete method to calculate it so far. If we need to calculate the solution of GP model, the method of iterative projection between two convex sets is useful. Let $d(A,B)$ is a Euclid distance between two point sets. If $d(A_0,B_0)=d(A_0,B)$, $B_0\in B$, then we call point B_0 is a projection from A_0 to set B. The method of iterative projection between two convex sets may calculate their shortest distance. For any $A_0\in A$, we seek $B_0\in B$ satisfying $d(A_0,B_0)=d(A_0,B)$. For B_0, we seek $A_1\in A$ satisfying $d(B_0,A_1)=d(B_0,A)$. For $A_i\in A$, we seek $B_i\in B$ satisfying $d(A_i,B_i)=d(A_i,B)$. For B_i, we seek $A_{i+1}\in A$ satisfying $d(B_i,A_{i+1})=d(B_i,A)$. When $d(A_i,B_i)<\varepsilon$, the iterative process stops and the calculation are completed. The convergence of the iterative process means $\lim_{i\to\infty} d(A_i,B_i)=d(A,B)=d(A^*,B^*)$.

About the process above we have the third theorem:

Theorem 3. If one of two convex sets A,B is boundary, the iterative process of the aforesaid alternative projection is convergent.

According to Theorem 3, seeking the distance between points can complete seeking the distance between two convex sets and a close convex set or subspace. So solving generalized linear model with convex constraint can be transformed into solving a common linear model with convex constraint many times. In the calculation example the convergence process is very fast.

Now we return to the SEM with multi-group evaluation. Suppose that each client will iterate N times until the iteration stops and will produce M coefficients, which will be regarded as observation variables. Suppose there are W groups to be measured. For each group we get a $N\times M$ data block. We accumulate these data block one by one and get a $WN\times M$ matrix together. The last calculate result is a W vector, whose $i'th$ component is the last evaluation value of $i'th$ group, $i=1,2,\mathrm{L}\ W$. The coefficients of the model reflect the effect among the variables.

5. COMPARISON AND CONCLUSIONS

In preceding paper and many other references, the researchers gave arbitrary initial value and the PLS method did not always ensure the convergence of iterative process [1]. It is reported that it may cost four or five minutes to adopt the PLS method in CSI model for 250 samples and 18 indicators. It is obviously too slow.

In this paper, we find a suitable iterative initial value for PLS algorithm in structural equation models, which enhances convergence rate greatly. For 250 samples it costs only 1 or 2 seconds to finish computing and the convergence rate has been enhanced by hundred of times.

This paper presents an improved PLS algorithm in SEM using a suitable iterative initial value with constraint of unit vector. The algorithm enhances the convergence rate greatly and its convergency is illuminated. Then we investigate multi-group SEM and adopt distributed computing to calculate all the coefficients in SEM of each group, which results in considerable timesavings when compared with traditional approach. Furthermore, we build a uniform model using the generalized linear model with convex constraint, and an algorithm for the multi-group SEM is established. The results of this paper have been received in software DASC.

Distributed computing in multi-group structural equation model has been realized and the programs are completed. As for space limitation, we have omitted the detail codes for this problem, but you can download from the on-line web site: http://public.whut.edu.cn /slx/english/index.htm.

6. REFERENCES

[1] C. Fornel, M. D. Johnson, et al, “The American Customer Satisfaction Index: Nature, Popurse, and Findings”, Journal of Marketing, October 1996, Vol.60: 7-18.

[2] Department of Quality Control, “General Administration of Quality Supervision and Inspection of the People's Republic of China”. China Centre for Enterprise Research, Tsinghua University, and Introduction to the Customer Satisfaction Index of China, Beijing: Standards Press of China, 2003.

[3] S.Y. Lee, H.T. Zhu, “Muximum likelihood estimation of nonlinear structural equation models”, Psychometrika, June 2002, 67(2): 189-210.

[4] S.Y. Lee, B. Lu, “Case-Deletion Diagnostics for Nonlinear Structural Equation Models”, Multivariate Behavioral Research, 2003, 38(3): 375-400.

[5] H.Q. Tong, “Evaluation Model and Its Iterative Algorithm by Alternating Projection”, Mathematical and Computer Modelling, 1993, 18(8): 55-60.

[6] H.Q. Tong, Data analysis & statistical computation (DASC) software, Beijing, Science Press, 2005.

Building Digital Library by Using P2P Techniques

Yuefeng Chen
Information College, Guangdong Ocean University,
Zhanjian, Guangdong Province, 524025,China
Email: yuefengch71@sina.com
Zhende Li, Yinqiao Peng , Lingxi Peng
Information College, Guangdong Ocean University,
Zhanjian, Guangdong Province, 524025,China
Email: dbface@21cn.com

ABSTRACT

The P2P (peer-to-peer) overlay network is a kind of distributed computing technology with many attractive features. In P2P architecture, each participant peer has autonomous faculty, which ensures that P2P technology can provide more flexible way to manage distributed resources than traditional client/server network. Digital library is a complex project, which aims to organize and integrate large-scale, distributed and various forms of data. We propose to build a feasible universal digital library utilizing hybrid P2P technology to improve system efficiency. The whole developing procedures are sequentially described, and the primary strategies, i.e. repository design and management, querying within a peer or peer group, creating peer group and register management, are briefly discussed.

Keywords*:* P2P, digital library, peer, distributed system.

1. INTRODUCTION

The rapid development of information technology, especially the emergence of Internet, has brought more convenience to human's life and work. Digital library will replace traditional library, providing people with more colorful and convenient services.

For digital library, various forms of distributed data (such as data, image, sound etc) and heterogeneous resources should be organized and integrated into a system, from which people can freely browse and handily download resources.

Digital library is a vast and complex information system, supported by technologies of information science, computer science and communication science. Many researchers and experts have devoted to designing and implementing digital library since 1995. And P2P technology has gained much attention in recent years. In this paper, we discuss the development of universal digital library based on P2P (we call that as DLBP).

The paper is organized as follows. Section 2 is an overall presentation of the digital library. Section 3 presents concept of the P2P technology and differences between P2P and traditional distributed systems. Section 4 discusses the processes and primary technologies of constructing digital library based on P2P. Finally, Section 5 is conclusion and prediction of future developments.

Yuefeng Chen : graduated from communication and information system speciality of south China University of technology, major in software engineering, communication and information engineering.

2. ARCHITECTURE OF DIGITAL LIBRARY

The Computer Science Technical Reports Project issues the architecture of standard digital libraries. It refers to some basic terms: digital object, repository, handle, and metadata [1]. Information stored in digital library is named as "digital objects", which mean not only collections of bits, but also structural and related information, such as intellectual property rights. Each digital object has a unique "handle" to identify, and repositories are wares where digital objects are stored and managed. The architecture essentially specifies those characteristics that apply to all types of material and provides a framework separated from the content stored in the library.

Four key components are included in digital library: Handle system, Repository, Searching system and User interface [2]. These components work together as following: the Handle system is responsible for creating digital objects; the Repository is responsible for storing and administering digital objects; the Searching system takes charge of information query; the User interface provides interfaces for users to communicate with system.

3. FEATURES OF P2P TECHNOLOGY

With direct exchange between peer nodes, P2P network can lighten bottleneck of central server and make effective use of computing and storage resources [3]. That means each node in P2P has the same role and responsibility, having a large degree of self-configuration and self-management properties. In P2P environment, the control is completely decentralized, and peers cooperate together by exploiting the locality of their interactions. In P2P, peers can be freely added to and deleted from network without any influence to others.

The applications of P2P technology can lighten bottleneck caused by central server, making effective use of computing and storage resources. Famous examples like Napster, Gnutella provide exchanging digital collections from any peer of network for end users to establish file-sharing network.

Usually, there are two kinds of P2P model: pure P2P and hybrid P2P. In the former, each peer is equal to cooperate. Yet the hybrid P2P contains servers to maintain and search lists of file system. As depicted in Fig. 1., the server (i. e. "Super peer ") connects some separate peers (for example, personal computer, or PDA etc.) into group according to their close positions. When a peer (PC 1) requires querying something, it firstly sends message to its server (step 1), and then the server returns it to the contented peer (PC 2). Thereby the peer (PC 1) directly establishes connection with PC 2 (step 2). The hybrid P2P is proved to have several advantages compared with pure P2P: better stability, faster

lookup, transparency and unblocked network [4].

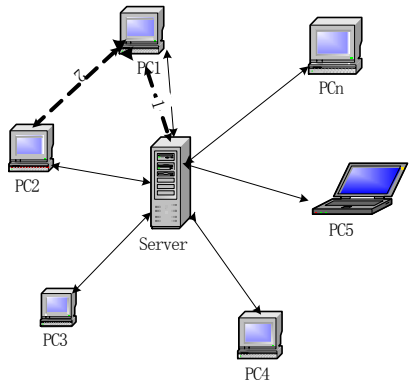


Fig.1. Hybrid P2P network

4. SYSTEM DEVELOPING

Digital libraries are large-scale information systems involved the whole procedure of processing information for various distributed resources.

4.1 System analysis

For P2P is helpful in developing distributed system, we propose to design the digital library utilizing P2P technology. Hybrid P2P technology is adopted for the objective system to improve unitary efficiency. In DLBP project, computers are grouped to several independent peer groups and each group takes charge of some related peers. We suggest classifying the peer groups according to position. The essential functions of each peer group are depicted in Fig. 2. as follows:

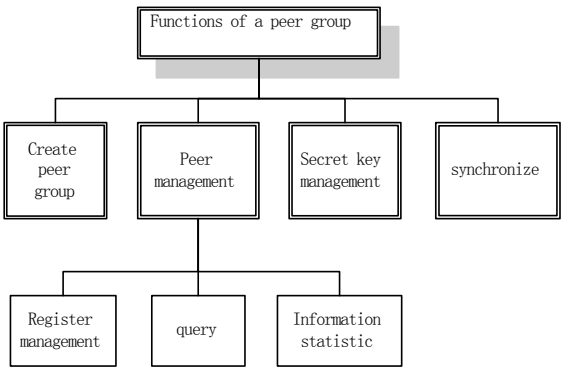


Fig. 2. Basic functions of a peer group

4.2 Primary strategies

The structure of digital library is highly dependent on the given type of information it contains and the accessing methods with which to retrieve information. We will discuss some primary techniques of strategies in constructing DLBP[5].

1)Repository design and management: A repository is a system for networked based storage and access to digital objects. A simple protocol named Repository Access Protocol (RAP) specifies interactions of repository. With RAP, we can write applications to manipulate repository via the open interface. When we design repository, the primary problem is security. To ensure that users should access collections safely, we keep digital object, its rights and permissions as an integral package in the repository. In addition, the implementation of every RAP command includes an explicit validation of the terms and conditions for access. Note that the internal organization of a repository is mostly independent of applications, which makes system flexible and maintainable. For example, there may be several versions of a digitalized photograph; they can be stored in the repository as a set of digital objects, identified by the handle.

2) Creating peer group*:* In DLBP, data are associated with keys defined over a virtual address space and each peer is responsible for a subset of the keys [6]. In order to improve system efficiency, peers are organized into autonomous groups with hierarchical distributed hash table (DHT). Thus, groups will first determine the group responsible for the key; in the same way, the selected group then uses its intra-group overlay to choose the specific peer that is responsible for the key. The critical problem of communication in a P2P community is associated with dynamic nature of peers. A simplified method is that participating computers can be grouped together logically as a multicast group. Guo [7] proposed DINCast to support group communications, which built on top of an existing P2P multicast tree and results in a hop efficient dynamic multicast infrastructure.

3) Register management: Before the participating peer is connected to network, its first behavior is initially introducing and joining. Two strategies are usually used in P2P environment: distributed indexing structures and routing based approaches. For a more detailed discussion see reference [8]. The latter that we will subsequently discuss is suitable to the digital library, which mostly contains plentiful multimedia objects. When constructing DLBP, summaries hosting the corresponding data objects are established on the unique peer or peer group. Once introduced into network, the peer will ask from others the required summaries to create an initial routing table and then communicate its own summary with other peers.

4)Querying: Information search from heterogeneous resources of digital library is a significant task, and academic search engine technology is faced challenges from commercial search engine tools(Google, Yahoo, etc). The detailed description of search engine technology see the paper [9]. Assumed that a reader issues a query in any registered peer (or group), that peer will determine what is the most promising peer (or groups) to advance according to the routing table. Thereafter the query is spread to others through the similar mechanism until the next peer is impossible to afford a useful result.

4.3 System implement

The open-source approach is better than the traditional closed-source approach because it offers a new and better method for developing high quality software. JXTA is an open-source and independent platform promoted by Sun Microsystems to establish a network programming platform for P2P systems. The core of JXTA is several protocols, for example, discovery protocol, membership protocol, pipe protocol, monitoring protocol and access protocol etc. Usually, we firstly build a small set of general and necessary infrastructures, and then assemble these components into higher-level services. For example, when constructing the DLBP, we initially implement a subset of the JXTA core protocols to build individual peer, which includes a given peer ID and a set of network interfaces (endpoints). Then we organize these separate peers implementing a common set of services into peer groups. We can similarly build other components and the whole system.

5. CONCLUSION

Building digital libraries will be related to the solutions of numerous scientific, technological, methodological, legal and other issues. The P2P has been proved to bring great efficiency in developing distributed system. In this paper, we propose to develop universal digital library based on hybrid P2P. The DLBP project will be more efficient in making better use of resources than client-server. We discussed primary strategies in this paper. And in the real work, sometimes we have to balance the storage cost, the network load, and the result quality with middle schemes. The further research should be focused on query optimizing and security of system.

6. REFERENCES

[1] Y. William, “Key concepts in the architecture of the digital library,” D-Lib Magazine, 1995,1(7).

[2] Y. William, “An architecture for information in digital libraries”. D-Lib Magazine, 1997,3(2).

[3] B. NADIA, M. ALBERTO, et al, ”Data-driven coordination in Peer-to-Peer information,” system. International Journal of Cooperative Information Systems, 2004, 13(1): 63-89.

[4] L. Garces, E. BIRSACK, et al, ”Hierarchical Peer-to –peer systems,” Parallel Processing Letters, 2003,13(4): 643-657.

[5] L.H.Wang, “Key techniques of digital libraries based on P2P”, journal of information, No q1, 2005, pp 24-28(in Chinese).

[6] Rowstron, P. D. Pastry, “Scalable, Distributed Object Location and Routing for Large- Scale Peer - to - Peer Systems”. Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg Germany, 2001.

[7] H.Q. Guo, L.H. Ngoh, et al, ”DINCast: a hop efficient dynamic multicast infrastructure for P2P computing”, Future Generation Computer Systems, 2005,21. 361–375.

[8] M. Wolfgang, E. Martin, et al, “Fast retrieval of high-dimensional feature vectors in P2P networks using compact peer data summaries,” Multimedia Systems, vol. 00, 2005. 1–11.

[9] L. Norbert, “Search engine technology and digital libraries,” D-Lib Magazine, 10(6), 2004.

Bootstrap Analysis of MOSFET Life Distribution with Negative Order Moment Estimate and Its Distributed Computing

QiaoLing Tong, XueCheng Zou
Department of Electronic Science and Technology, Huazhong University of Science and Technology
Wuhan, Hubei, 430074, P.R.China
Email: qltong@whicc.com
HengQing Tong
Department of Mathematics, Wuhan University of Technology
Wuhan, Hubei, 430070, P.R.China
Email: hqtong2005@pubulic.whut.edu.cn

ABSTRACT

In this paper, a MOSFET life model is proposed by making use of the threshold voltage drift property. The negative order moment estimator is introduced and it is the maximum likelihood estimator. A new method of parameter estimate, F estimate of parameter, is proposed for truncated testing. Bootstrap re-sample method is used to determine the interval of the parameter by F estimate as the sample size is small in truncated test and the bootstrap control chart which is agile and robust enough can be drawn out. This kind of bootstrap analysis can be realized by distributed computing through the website.

Keywords: MOSFET Life Distribution, Negative Order Moment Estimator, F Estimate, Distributed Computing.

1. Introduction

In deep-sub-micron MOSFET devices, lifetime has been greatly influenced by hot carrier degradation [1]. So it is necessary to build an accurate MOSFET life model for the device reliability. In semiconductor manufacture system, the sample size is small because of the time and cost limit. Bootstrap is a stable and accurate statistical method in this situation [2]. This method is introduced by professor Efron in 1979 and has been used in many manufacture systems [3]. In this paper, a new MOSFET life model is proposed and F estimate based on bootstrap method is used to analyze this model.

2. MOSFET LIFE MODEL BY THRESHOLD DRIFT PROPERTY AND ITS DISTRIBUTION

The MOSFET degradation has a strong impact on its performance. Experiments have shown that we can deduce the level of the MOSFET degradation according to the analysis of the parameter ΔVth. When $\Delta Vth > (\Delta Vth)_{\max}$, the MOSFET cannot work normally [4,5].

The value of the ΔVth increases as the work time of a MOSFET increases. In general, we suppose

$$\Delta V_{th} = A t^{n} \tag{1}$$

Where A denotes the change rate of ΔVth. It may be affected by many independent little factors such as Vds, tox, L and so on. n is a constant and it is different between nMOSFET and pMOSFET. We suppose that A is a random variable with normal distribution with definition $A > 0$:

$$f(A) = \begin{cases} \dfrac{C}{\sqrt{2\pi}\,\sigma} \exp\left(-\dfrac{(A-\bar{A})^2}{2\sigma^2} \right), & A > 0 \\ 0, & A \le 0 \end{cases} \tag{2}$$

From $\int f(A)dA = 1$, we know $C = \dfrac{1}{\Phi(\bar{A}/\sigma)}$, where Φ is the distribution function of standard normal.

Let T be the MOSFET life, $T = \left((\Delta V_{th})_{\max} \Big/ A \right)^{1/n}$, then T is also a random variable. The density function of MOSFET life distribution is

$$f(t) = \begin{cases} \dfrac{n(\Delta V_{th})_{\max}}{\sqrt{2\pi}\,\sigma\Phi(\bar{r}/\sigma)t^{n+1}} \exp\left(-\dfrac{((\Delta V_{th})_{\max} - \bar{A}t^n)^2}{2\sigma^2 t^{2n}} \right), & t > 0 \\ 0, & t \le 0 \end{cases} \tag{3}$$

Let $(\Delta V_{th})_{\max} \Big/ \sigma = a$, $\bar{A} \Big/ \sigma = b$, then the MOSFET life distribution density function is

$$f(t) = \begin{cases} \dfrac{na}{\sqrt{2\pi}\ \Phi(b)t^{n+1}} \exp\left(-\dfrac{1}{2}\left(\dfrac{a}{t^n} - b \right)^2 \right), & t > 0 \\ 0, & t \le 0 \end{cases} \tag{4}$$

And the distribution function is

$$F(t) = \frac{1 - \Phi(\frac{a}{t^n} - b)}{\Phi(b)} \tag{5}$$

3. NEGATIVE ORDER MOMENT ESTIMATOR OF COMPLETE SAMPLE

In Eq. (3), $f(t)$ has three parameters $(\Delta V_{th})_{\max}$, σ, $\bar{A}$. They are independent on the physical process, but they are dependent on the density function. We can estimate a, b from Eq. (4). We need to estimate $(\Delta V_{th})_{\max}$ in addition, and then we can estimate σ, $\bar{A}$.

First, we consider moment estimator. For common moment estimator Et^j, $j = 1, 2, \cdots$, we cannot calculate its integration. If we make use of negative order moment estimator Et^j, $j = -1, -2, \cdots$, the integration can be calculated easily.

$$E(t^{-n}) = \int_0^{+\infty} \frac{na}{\sqrt{2\pi}\ \Phi(b)t^{2n+1}} \exp\left(-\frac{1}{2}\left(\frac{a}{t^n} - b \right)^2 \right) dt$$

$$= \frac{1}{a\Phi(b)} \int_{-b}^{+\infty} (y+b) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy \quad y = \frac{a}{t^n} - b$$

$$= \frac{b}{a} + \frac{1}{a\sqrt{2\pi}\Phi(b)} \exp\left(-\frac{1}{2}b^2\right) \tag{6}$$

$$E(t^{-2n}) = \int_0^{+\infty} \frac{na}{\sqrt{2\pi}\ \Phi(b) t^{3n+1}} \exp\left(-\frac{1}{2}\left(\frac{a}{t^n} - b\right)^2\right) dt$$

$$= \frac{1}{a^3\Phi(b)} \int_{-b}^{+\infty} (y+b)^2 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy \quad y = \frac{a}{t^n} - b$$

$$= \frac{b^2+1}{a^2} + \frac{1}{a^2\sqrt{2\pi}\Phi(b)} \exp\left(-\frac{1}{2}b^2\right) \tag{7}$$

Let $t_1, \ldots, t_m$ be $i.i.d.$ sample data of the MOSFET life. We make use of the sample moment to estimate the population moment. The difference is the sign of the order of moment. From the system of equations,

$$s_1 = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{t_i^{\,n}} = \frac{b}{a} + \frac{1}{a\sqrt{2\pi}\Phi(b)} \exp\left(-\frac{1}{2}b^2\right) \tag{8}$$

$$s_2 = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{t_i^{2n}} = \frac{b^2+1}{a^2} + \frac{1}{a^2\sqrt{2\pi}\Phi(b)} \exp\left(-\frac{1}{2}b^2\right) \tag{9}$$

we have

$$b = \frac{1}{s_1}\left(as_2 - \frac{1}{a}\right) \tag{10}$$

$$a = \frac{1}{s_1}\left[b + \frac{1}{\sqrt{2\pi}\Phi(b)} \exp\left(-\frac{1}{2}b^2\right)\right] \tag{11}$$

Moreover, we can obtain the solution $\hat{a}$ and $\hat{b}$ from Eq. (10) and Eq. (11). They are the moment estimate of a and b.

Second, we consider the maximum likelihood estimator of parameters. The likelihood function is

$$L(a,b) = \left[\frac{a}{\sqrt{2\pi}\Phi(b)}\right]^n \frac{1}{\prod_{i=1}^{m} t_i^{\,n+1}} \exp\left[-\frac{1}{2}\sum_{i=1}^{m}\left(\frac{a}{t_i^{\,n}} - b\right)^2\right] \tag{12}$$

Taking derivation of $L(a,b)$, we have

$$\frac{\partial L}{\partial a} = n\left[\frac{a}{\sqrt{2\pi}\Phi(b)}\right]^{n-1} \frac{1}{\sqrt{2\pi}\Phi(b)} \cdot \frac{1}{\prod_{i=1}^{m} t_i^{\,n+1}} \exp\left[-\frac{1}{2}\sum_{i=1}^{m}\left(\frac{a}{t_i^{\,n}} - b\right)^2\right] + \left[\frac{a}{\sqrt{2\pi}\Phi(b)}\right]^n \frac{1}{\prod_{i=1}^{m} t_i^{\,n+1}} \cdot \exp\left[-\frac{1}{2}\sum_{i=1}^{m}\left(\frac{a}{t_i^{\,n}} - b\right)^2\right]\left[-\sum_{i=1}^{m}\left(\frac{a}{t_i^{\,n}} - b\right)\frac{1}{t_i^{\,n}}\right] = 0 \tag{13}$$

$$\frac{\partial L}{\partial b} = n\left[\frac{a}{\sqrt{2\pi}\Phi(b)}\right]^{n-1} \frac{a\Phi'(b)}{\sqrt{2\pi}\Phi^2(b)} \cdot \frac{1}{\prod_{i=1}^{m} t_i^{\,n+1}} \exp\left[-\frac{1}{2}\sum_{i=1}^{m}\left(\frac{a}{t_i^{\,n}} - b\right)^2\right] + \left[\frac{a}{\sqrt{2\pi}\Phi(b)}\right]^n \frac{1}{\prod_{i=1}^{m} t_i^{\,n+1}} \cdot \exp\left[-\frac{1}{2}\sum_{i=1}^{m}\left(\frac{a}{t_i^{\,n}} - b\right)^2\right]\left[\sum_{i=1}^{m}\left(\frac{a}{t_i^{\,n}} - b\right)\right] = 0 \tag{14}$$

Simplifying Eq. (13) and Eq.(14), we have

$$a^2 s_2 - ab s_1 = 1 \tag{15}$$

$$a s_1 = b + \frac{\Phi'(b)}{\Phi(b)} \tag{16}$$

Noticing that Eq. (11) and Eq. (15), we know that the moment estimator with negative order of MOSFET life distribution is just its maximum likelihood estimator. It is more reasonable to take the negative order moment estimator for MOSFET life distribution.

4. PARAMETER ESTIMATE FOR TRUNCATED TESTING

Life testing is often truncated by the size of samples, because the time and the cost of testing are limited. Suppose m products are taken as life testing. We have observed M products, which have been failures. $t_1 \le t_2 \le \cdots \le t_M$ are M proceeding order statistics of products life. For exponential distribution, Weibull distribution, normal and lognormal distributions, we can give parameter estimators in truncated testing. These distributions can be transformed into standard distributions without any parameter. According to the order statistics of the standard distribution, we can obtain the Best Linear Unbiased Estimation by the least square method. But we cannot do it in this way for MOSFET life distribution because we cannot transform it into any standard distribution without any parameter. The maximum likelihood estimate of MOSFET life distribution from the order statistics of the truncated samples is not certain to converge, because its density function does not satisfy some of the convergence conditions.

In this paper, a statistical method, F estimate of parameter, is proposed. It is suitable to any distribution whether or not it can be transformed into a standard distribution. What is more, it has no definition of sample size.

We can consider the problem in this way. For the lives of m products in life testing, we have observed M preceding lives of products. We have not observed $m - M$ late lives of products, but they are existent. For the empirical distribution function of life t, we have observed $F_m(t_1) = 1/n, \ldots, F_m(t_M) = M/m$. We have not observed the late values, but they are existent too. From Glivenko-Cantelli Theorem, we know that the empirical distribution function of random variable with probability one converges to its distribution function uniformly. That is,

$$P\left\{\lim_{n\to\infty}\left(\sup_{-\infty<t<\infty}\left|F_n(t) - F(t)\right|\right) = 0\right\} = 1 \tag{17}$$

Therefore, we can obtain M approximate equations

$$F(t_1) = \frac{1}{m}$$
$$F(t_2) = \frac{2}{m}$$
$$\vdots$$
$$F(t_M) = \frac{M}{m} \qquad (18)$$

The parameters to be estimated are contained in the system of Eq. (18). The problem is changed into solving a nonlinear regression model. Solving the model by the least square method, we can obtain the parameters to be estimated. We call it F estimate of parameter. Obviously, this method is reliable in theory, simple in computation and extensive in application.

From Eq. (5), we know the MOSFET life distribution. The regression equations are

$$\frac{1-\Phi(\frac{a}{t^n}-b)}{\Phi(b)} = \frac{i}{m}, \qquad i = 1,\ldots,M \qquad (19)$$

Making use of the least square method, we need to seek the minimum by changing parameters a , b :

$$Q(a,b) = \sum_{i=1}^{M}\left[\frac{1-\Phi(\frac{a}{t^n}-b)}{\Phi(b)} - \frac{i}{m}\right]^2 \rightarrow \min \qquad (20)$$

5. BOOTSTRAP ANALYSIS FOR F ESTIMATION AND ITS DISTRIBUTED COMPUTING

In order to investigate the properties of MOSFET life distribution carefully and also in order to test the statistical method, F estimate of parameter, we need to take simulative computation by Monte Carlo method. First, we generate $m = 200$ random numbers according to MOSFET life distribution density function Eq. (4), suppose that the parameters a=1, b=1, n=0.841. Then we put these n samples into Eq.(20) and use Powell algorithm improved by Sargent to solve it. We find that when samples are completed the result is very satisfied: $a = 0.901$, $b = 0.947$, $n = 0.688$.

When we truncated the samples by number $M = 100$, that is we take the first M random samples, the fitting result is not so satisfied.

The 100 samples of MOSFET life are in Table 1.:

Table 1. MOSFET Life t ($M = 100$, unit: hour)

0.042	0.027	0.036	0.036	0.039
0.087	0.090	0.099	0.099	0.042
0.174	0.180	0.180	0.186	0.099
0.240	0.240	0.243	0.243	0.108
0.300	0.300	0.306	0.312	0.192
0.381	0.396	0.411	0.417	0.195
0.465	0.471	0.483	0.486	0.243
0.516	0.528	0.537	0.537	0.258
0.417	0.048	0.048	0.054	0.054
0.420	0.138	0.141	0.144	0.165
0.489	0.195	0.207	0.222	0.240
0.489	0.261	0.261	0.270	0.297
0.540	0.339	0.342	0.354	0.363
0.540	0.423	0.438	0.438	0.444
0.618	0.498	0.498	0.501	0.504
0.624	0.552	0.558	0.564	0.570
0.699	0.636	0.639	0.642	0.660
0.699	0.705	0.714	0.717	0.741
0.417	0.048	0.048	0.054	0.054
0.420	0.138	0.141	0.144	0.165

The estimated parameters are: $a = 1.66$, $b = 0.59$, $n = 0.243$. The results are presented in Fig. 1..

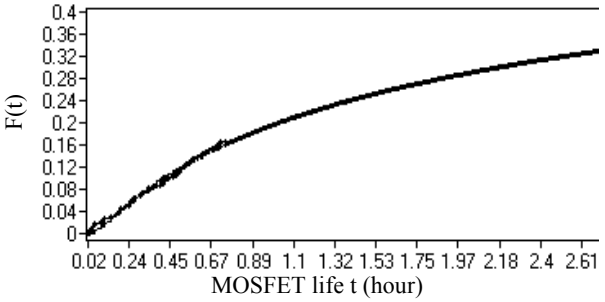


Fig. 1. The F (t) figure as truncated samples

We can improve the result of truncated samples by bootstrap method. Suppose $t_1 < t_2 < \ldots < t_M$ and we construct an empirical distribution function: $F_m^* = \frac{k}{m}$, when $0 < t < t_M$. And F_m^* is unknown when $t > t_M$. We resample only for $0 < t < t_M$. For each re-sample $t_1^*, t_2^*, \ldots, t_M^*$, we can calculate the parameters estimation a, b, n. We resample J times and obtain a series parameters estimations a_i, b_i, n_i , $i = 1,2,\ldots,J$. When J is sufficient large, we can get the intervals of parameters estimation a, b, n. For fiver confidence level α , the lower limit of interval estimate of a is a_L , here L satisfies $L = J \cdot \alpha$, the upper limit of interval estimate of a is a_U , here U satisfies $U = J \cdot (1-\alpha)$. Similarly we can get the interval estimate of parameters $b, \lambda.$

The products with MOSFET are popular everywhere, billions of chip are working at a time, so the MOSFET life test can be taken everywhere. One chip contains thousands of MOSFETs, if one MOSFET is damaged, the whole chip cannot work at all. If we simplify the problem of the products life and suppose m chips are working in a unit, M chips are damaged after sometime, then we can deduce the MOSFET life distribution through the method proposed in this paper and don't need all the chips are damaged. There are many units and we can use the distributed computing. Many units report their parameters estimate of MOSFET life distribution to center computer; we can obtain a series parameters estimations a_i, b_i, n_i , $i = 1,2,\ldots,J$. When J is sufficient large, we can get the intervals of parameter estimations a_i, b_i, n_i , and obtain the distribution of the whole product. The distributed computing can be done through website. We compile the program by JAVA language which can find the computer by its IP address and send the subprogram to it.

6. Appendix

We can use following main code for distributed computing with RMI, and only substitute the different subfunction. The detailed program code can be downloaded from our page site: http://public.whut.edu.cn/slx/english/index.htm.

```
public interface TestInterface extends
Remote
{
public int Simple(int n,int m)
throws java.rmi.RemoteException;
public int[ ][ ] Comp(int n,int s,int t)
throws java.rmi.RemoteException;
} public class TestImpl extends
UnicastRemoteGROUP implements
TestInterface
{
public int Simple(int n,int m)
throws java.rmi.RemoteException
{ return n/m+1; }
public int[ ][ ] Comp(int n,int s,int t)
throws java.rmi.RemoteException
{
//This is the computing algorithm
}
}
public class TestServer
{
public TestServer()
{
try{
TestInterface c=new TestImpl( );
Naming.rebind("//192.168.1.101/TestRMI",c);
}catch(Exception e){
e.printStackTrace(); }
}
public static void main(String agrs[ ])
{ new TestServer( ); }
}
public class TestClient1
{
try{
TestInterface c=(TestInterface)
Naming.lookup("//192.168.1.101/TestRMI");
int[ ][ ] result=
c.Compute(1000,1000/186,1000/147+1);
}catch(Exception e){
e.printStackTrace( ) ;}
}
```

7. References

[1] Ming Hsiang Chiou, Klaus Y. J. Hsu, “Wideband modeling technique for deep sub-micron MOSFETs", Solid-State Electronics, 2004, 48(10-11): 1891-1896.

[2] W. Zhang, Q.N. Wang, “Bootstrap Control Charts”, Quality Engineering, 1996,9(4): 143-150.

[3] B. Efon, “Bootstrap Methods: Another look at the Jackknife”, Ann. Stat., 1979, Vol.7: 1-36.

[4] Nuditha Vibhavie Amarasinghe, Anna Zlotnicka, Fang Wang, “Model for random telegraph signals in sub-micron MOSFETS”, Solid-State Electronics, 2003, 47(9): 1443-1449.

[5] Ren Chuen Chen et al, “An iterative method for adaptive finite element solutions of an energy transport model of semiconductor devices”, Journal of Computational Physics, 2003, 189(2): 579-606.

Distributed Computing For Modeling The Decomposition Products Of A Protein

Qiaoling Tong
Department of Electronic Science and Technology, Huazhong University of Science and Technology
Wuhan, Hubei, 430074, China
Email: qltong@whicc.com
And
YanFang Deng, Zhao Xu
Department of Mathematics, Wuhan University of Technology
Wuhan, Hubei, 430070, China
Email: hqtong2005@public.whut.edu.cn

ABSTRACT

The molecular weight of a protein may be partitioned in different ways as the sum of the molecular weights of its constituting proteins. The problem of enumerating the different decomposition products is equivalent to that of finding different partitions of an integer. In order to solve this kind of problem, this article firstly gives an ingenious and succinct decomposition algorithm without the help of a computer, and draws a conclusion that there existing a positive integer X_0 , X can be decomposed for any given $X \geq X_0$. Then in the situation which the computer assists, this article gives another decomposition algorithm demonstrating that with the increase of the molecular weight of protein, the solution number and time are increasing constantly too, and at the same time it exists *NP* difficulty. In order to solve all the above problems, this article provides improved algorithm, and introduces the distributional thought which not only helps avoiding *NP* difficulty, moreover also greatly reduces the solution time.

Keywords: Decomposition Products of a Protein, an Integer Decomposition Problem, Distributional Thought, *NP* Difficulty.

1. INTRODUCTION

Protein is an essential ingredient which constitutes all cells and organizations; it is the most important material base of humanity activities. It is a high polymer, which is mainly formed by more than 20 kinds of amino acid. Because the amino acid member arrangement and the combination way are different, it will have many kinds of combinations of amino acids for a given protein [1]. In fact, under the situation that the molecular weight X of a protein and the molecular weights $a(i)$ $(i=1, 2 \ldots n)$ of n kinds of amino acids are known, we can turn to solve all integer solutions of the following linear equation

$$\sum_{i=1}^{n} a(i)x(i) = X, x(i) \geq 0 \qquad (1)$$

Usually, we use exhaustive searching method to complete the computing process that how many kinds of the combinations of amino acids a given protein can be decomposed into [2]. All the solutions can be gained in theory. Actually, the number of the combinations is $\prod_{i=1}^{n}\left[X/a(i)+1\right]$ according to estimation [3]. Since the constitution types n of a protein are so many (The types in existence we have known are more than 20), and the molecular weight X is large, the computing time and the complexity of computation will be very big, moreover, it exists *NP* difficulty [4]. In this situation, it would be impossible for any one computer or person to solve such a big task in a reasonable amount of time; therefore, we introduce the thought of distributed computing [5].

Distributed computing is a science which solves a large problem by giving small parts of the problem to many computers to solve and then combining the solutions for the parts into a solution for the problem. It mainly uses the unused handling ability of computers' CPU on the Internet to solve the large-scale estimation problem.

Along with the universal application of the computer network, the requirements to make use of distributed computing basing on it become more and more intense. Any computer in the network can be realized as a main node computer to carry out the computation of the main process, while the other computers will achieve distributed computing in its guidance for the calculation. The more computers the network has, the more powerful it would be.

2. AN INGENIOUS AND SUCCINCT DECOMPOSITION ALGORITHM

We can treat the process of decomposing a protein into amino acids as the process of decomposing a positive integer into n positive integers' linear combination. During this process, the positive integer X (the molecular weight of the protein) and $a(i)$, $i=1, 2 \ldots n$ (the molecular weights of the acids) are known, $a(1) < a(2) < a(3) < \mathrm{K} < a(n)$. We need to find integers $x(i) \geq 0 (i = 1,2,\mathrm{K}\ n)$ such that

$$X = a(1)x(1) + a(2)x(2) + \Lambda + a(n)x(n) \qquad (2)$$

For example, n=18, $\{a(i)\}$ is

57, 71, 87, 97, 99, 101, 103, 113, 114, 115,
128, 129, 131, 137, 147, 156, 163, 186

If there exists $x(i), i = 1,2,\mathrm{K}\ ,n$, we say X can be partitioned and call $x(i), i = 1,2,\mathrm{K}\ ,n$ is a group of combination coefficient of X. We can consider the question above as *NP*. For problem of this kind, the first problem we need to resolve is whether there exists a positive integer X_0 which satisfies the condition that for each integer $X > X_0$, X can be decomposed. In order to find a

solution to the problem, we introduce the following algorithm:

Step 1 $\Delta(i) = a(i+1) - a(i), i = 1,2,\mathrm{K}\ ,n-1$**,** we get a number array as follows:

14,16,10,2,2,2,10,1,1,13,1,2,6,10,9,7,23

Step 2 Select proper $\{\Delta(i_j)\}$ from set $\{\Delta(i)\}(j = 1,2,\mathrm{K}\ ,n-1)$ to construct a geometry sequence $\{G(i)\}$:

1, 2, 4, 8, 16, 32

Example:

$$\Delta(8) = 1, \Delta(4) = 2\Delta(5) + \Delta(6) = \Delta(4) + \Delta(5) = 4\Delta(4) + \Delta(13)$$
$$= 8\Delta(2) = 16\Delta(13) + \Delta(14) + \Delta(15) + \Delta(16) = 32$$

Step 3 Based on the relation between the selected $\{\Delta(i_j)\}$ and $\{a(i)\}$, choose a group of radix elements $\{a(i_j)\}$ from $\{a(i)\}$:

$$a_2 = 71, a_4 = 97, a_8 = 113, a_{13} = 131$$

The sum of these numbers is $X_0 = 412$.

Step 4 It is easily proved that any integer $X_1 < a(1) = 57$ can be composed of $\{\Delta(i_j)\}$.

Step 5 For every integer $X > X_0$

$$X = a(1) * t + \sum a(i_j) + X_1 \quad (0 < X_1 < a(1)) \qquad (3)$$

where $t \geq 0, \sum a(i_j) = X_0$.

Step 6 Because $\sum \Delta(i_j) = X_1$, we can get the result:

$$X = a(1) * t + \sum a(i_j) + X_1$$
$$= a(1) * t + \sum a(i_j) + \sum \Delta(i_j)$$
$$= a(1) * t + \sum a(i_j + h) \qquad (4)$$

where $h \geq 0$. Example:

X=467=412+55=71+97+113+131+55
=71+97+113+131+ (32+16+4+2+1)
=71+97+113+131+ [(163-131) + (87-71)
+ (103-99) + (99-97) + (114-113)]
=163+87+103+114

In this algorithm, the decomposed coefficient can be different due to the varied choices of $\{\Delta(i_j)\}$ and $\{a(i_j)\}$. The value of X_0 can be reduced by observing and analyzing the relation among $\{\Delta(i)\}$. Finally we obtain the best elements $\{a(i_j)\}$:

$$a_4 = 97, a_8 = 113$$

The sum of them is $X_0 = 307$.

Correspondingly, $\{\Delta(i_j)\}$ are:

$$\Delta(8) = 1, \Delta(4) = 2, \Delta(4) + \Delta(5) = 4,$$
$$\Delta(4) + \Delta(5) + \Delta(6) = 6, 2(\Delta(4) + \Delta(5)) = 8,$$
$$2(\Delta(4) + \Delta(5)) + \Delta(6) = 10,$$
$$2(\Delta(4) + \Delta(5) + \Delta(6)) = 12,$$
$$\Delta(4) + \Delta(5) + \Delta(6) + \Delta(7) = 16,$$
$$\Delta(4) + \Delta(5) + \mathrm{K}\ + \Delta(11) = 32.$$

Now we are able to prove that the given X can be decomposed where $X \geq X_0$.

We will adopt mathematical induction to testify the aforementioned conclusion.

Proof: First, $X = X_0 = 307 = 97 + 97 + 113$

Suppose $X_k = X_0 + K$ can be decomposed, that is to say, we can decompose X_k as follow:

$$X_k = a(1) * t + \sum a(i_j) + X_1$$
$$= a(1) * t + \sum a(i_j) + \sum \Delta(i_j)$$
$$= a(1) * t + \sum a(i_j + h) \qquad (5)$$

And then when $X_{k+1} = X_0 + K + 1$

$$X_{k+1} = a(1) * t + \sum a(i_j) + X_1 + 1$$
$$= a(1) * t + \sum a(i_j) + \sum \Delta(i_j) + 1$$
$$= a(1) * t + \sum a(i_j) + \sum \Delta(i_j) + \Delta(1)$$
$$= a(1) * t + \sum a(i_j) + \sum \Delta^*(i_j)$$
$$= a(1) * t + \sum a(i_j + m) \qquad (6)$$

where $m \geq 0$, the procedure indicates that $X_{k+1} = X_0 + K + 1$ can also be decomposed.

It is well supported by the algorithm introduced above that there certainly exists a positive integer $X_0 = 307$ that for any given $X \geq X_0$, X can be decomposed. The advantage of this algorithm is that it can decompose an integer X quickly and briefly without the assistant of a computer when the integer X is small. However, when X is very big, the algorithm becomes complexity and inefficient, what's worse is that we can hardly get all of the possible solutions.

3. AN IMPROVED ENUMERATIVE ALGORITHM ON COMPUTER

In order to obtain all decomposition solutions to the above problem, we can use the branch and delimitation law in virtue of computer programming. During the process of computation, we get the experimental data in Table 1.:

Table 1.

X	The number of combinations	Running time(second)
200	4	0.4
300	14	1
400	45	2
500	158	6
600	522	14
700	1508	52
800	4291	130
900	11249	341
1000	28286	845

From the table above, we can observe that the number of combinations and the running time are unceasingly increasing with the increment of the molecular weight. Specially, when $X = 1000$, the combinations come to 28286 and the running time reaches 845 seconds which is such a long time for our computer to compute. While the fact is that the molecular weight of a protein may achieve several ten thousand or even more than that, so the method above will not be feasible in practice. In order to reduce the computing time, we must improve our algorithm.

This is a mathematical model, we propose the following

suppositions:

a. The molecular of an amino acid can not be further decomposed.
b. Regardless of the arrangement order of amino acids.
c. Each kind of amino acid in the protein is in existence with equal probability.

According to the hypotheses above, we fetch in another algorithm as follows:

Step 1 First compute $M_1 = {}^{X}\!/_{a(n)}$ and $M_2 = {}^{X}\!/_{a(1)}$. Supposing for any given positive integer *M*, we have $M_1 < M < M_2$.

Step 2 Make use of the standard subprogram *NEXCOM* (generating all order n-partition of *M*) to partition *M* as

$$M = x(1) + x(2) + \Lambda + x(n) \qquad (7)$$

Step 3 For $x(i), i = 1,2,\mathrm{K}\ ,n$, compute

$$Y = a(1)x(1) + a(2)x(2) + \Lambda + a(n)x(n) \qquad (8)$$

Step 4 If $Y = X$, then the integer sequence $x(i), i = 1,2,\mathrm{K}\ ,n$ is a group of combination coefficient of *X*.

Step 5 If $Y \neq X$, then continue making use of *NEXCOM* to partition *M* until we get all groups of combination coefficient $x(i), i = 1,2,\mathrm{K}\ ,n$, and utilize the Eq.(8) to compute *Y*. If there is not $x(i)$ which can satisfy $Y = X$, go to Step 6.

Step 6 $M = M + 1$, go to Step 1.

Step 7 If there is not $x(i)$ which can satisfy Eq.(8), we can say that Y can't be decomposed.

Example: The results of partition for $X = 305$ are giving in Table 2., $X = 306$ can not be partitioned:

Table 2. X = 305

57	71	87	97	99	101	103	113	114
0	0	0	0	0	2	0	0	0
0	0	0	0	1	0	2	0	0
0	0	1	0	0	0	1	0	0
0	0	3	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0
0	2	0	0	0	0	0	0	0

115	128	129	131	137	147	156	163	186
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0

In addition, we get some experimental data in Table 3. as follows by computer programming:

Table 3.

X	The number of combinations	Running time(second)
200	4	<1
300	14	<1
400	45	<1
500	158	2
600	522	3
700	1508	6
800	4291	14
900	11249	34
1000	28286	94

The advantage of this algorithm lies in its efficiency to obtain all the solutions, while at the same time it exists some disadvantages, for instance, when the molecular weight of a protein is very large, the value scope of *M* will be still very large, the computing time needed is also very considerable if using this algorithm, what's worse, it also exists *NP* difficulty. At this moment, if we make use of the thought of distributed computing, the time required will be greatly reduced, and the difficulty will be effectively resolved.

4. DISTRIBUTED COMPUTING

RMI (Remote Method Invocation) provides a simple and direct way for distributed computing. In this system, each computer can mutual access the other computers' functions and methods, and start their processes. The simplest system contains a server and several clients. The role of the server is to allot and store data. More important, it serves as the commander of this system, the other computers works in its guidance. All the algorithms and functions are carried out on the server. On the other hand, all the clients can invoke the algorithms and functions to compute at the same time. Finally, when all the computations are finished, the results will be sent back to the server by the client computers.

The concrete method adopting RMI to realize distributed computing for this problem is as follows:.

Step 1 There exists a protein molecular on the server, whose molecular weight is *X*, compute $M_1 = {}^{X}\!/_{a(n)}$ and $M_2 = {}^{X}\!/_{a(1)}$, the value scope of M is $M_1 < M < M_2$, let $M^* = M_2 - M_1$.

Step 2 Distribute the $M^* + 1$ positive integers within the range of $[M_1, M_2]$ to all the clients averagely, and then make them search the results independently according to the algorithm above in order to get all the solutions $x(i), i = 1,2,\mathrm{K}\ ,n$.

Step 3 Send all the results on the clients to the server. After the server's collection and analysis, we get all the possible combinations of amino acids that this protein decomposed into.

According to the foregoing method, we obtain the following data by applying six computers in the same LAN:

Table 4.

X	The number of combinations	Running time(second)
200	4	<1
400	45	<1
600	522	<1
800	4291	2
1000	28286	13
1100	67339	25
1200	154143	41
1300	338158	120
1400	716481	260

1500	1467221	324

In the process of realizing the distributed computing above each computer carries on the computation independently at the same time, therefore, the time spent in searching is greatly reduced, besides, it is easily found that the more computers used in the computing, the less runtime it will take.

Additionally, we could bring forward some assumptions below in practice:

a. If the total number of amino acids in a certain protein is known (get from apparatus), we can predigest Eq.(1) into the following form:

$$\begin{cases} \sum_{i=1}^{n} a(i)x(i) = X \\ x(i) \geq 0 \\ \sum_{i=1}^{n} x(i) = q \end{cases} \tag{9}$$

where q is the observed number of amino acids.

b. In fact, in the process of a protein's first-degree structure mensuration, we usually can make use of the mixed liquid gained through the protein's adequate hydrolyzation to make the ionic exchange chromatographic analysis, paper chromatographic analysis or thin layer chromatographic analysis. Through this test, we may determine some several kinds of amino acids contained in this protein. Then the Eq.(1) can be predigested as follows:

$$\begin{cases} \sum_{i=1}^{n} a(i)x(i) = X' \\ x(i) \geq 0 \end{cases} \tag{10}$$

where X' is the surplus molecular weight that X subtracts the sum of the molecular weight of the known amino acids. Since X is reduced, the searching scope and time is also greatly reduced. In the situation that the two hypotheses do exist separately, the computation quantity and searching time will be further reduced by virtue of the algorithm and distributed computing above.

5. CONCLUSION

By analyzing and researching protein's amino acids combination problem, we discover that we can solve some difficult problems such as decomposing any given positive integer into $\{a(i)\}$ which has definite dimensions, or decomposing any given positive integer into $\{a(i)\}$ which has an alterable dimensions by making use of the thought of distributed computing to solve the integer decomposition problem, moreover, the *NP* difficulty can be avoided effectually.

6. APPENDIX

As for space limitation, we have omitted the detail codes for this problem, but you can download from the on-line web site: http://public.whut.edu.cn /slx/english/index.htm. We give the main code for distributed computing with RMI as follows:

```
public interface TestInterface extends
      Remote
{
  public int Simple (int n, int m)
      throws java.rmi.RemoteException;
  public int[][] Comp(int n,int s,int t)
      throws java.rmi.RemoteException;
}
public class TestImpl extends
    UnicastRemoteGROUP implements
    TestInterface
{
  public int Simple(int n,int m)
      throws java.rmi.RemoteException
  {   return n/m+1;   }
  public int[][] Comp(int n,int s,int t)
    throws java.rmi.RemoteException
  {
    //This is the computing algorithm
  }
  }
  public class TestServer
  {
  public TestServer()
  {
    try{
        TestInterface c=new TestImpl();
         Naming.rebind("//192.168.1.101/TestRMI",c);
    }catch(Exception e){
        e.printStackTrace(); }
  }
  public static void main(String agrs[])
    { new TestServer(); }
  }
  public class TestClient1
  {
  try{
      TestInterface c=(TestInterface)
        Naming.lookup("//192.168.1.101/TestRMI");
      int[][] result=
              c.Compute(1000,1000/186,1000/147+1);
      }catch(Exception e){
        e.printStackTrace() ;}
  }
```

7. REFERENCES

[1] Q. Ruan, H. Lu, et al, Protein's Amino acids Combination Problem, Beijing: Academic Press of Beijing Union University, 1994.

[2] H.Q. Tong, "Modelling the Decomposition Products of a Protein", Mathematical and Computer Modelling, 1995, 20(9): 45-50.

[3] D.I.A. Cohen, Basic techniques of combinatorial theory, John Wiley & Sons, 1978.

[4] I.P. Goulden, D.M. Jackson, Combinational enumeration, John Wiley & Sons, 1983.

[5] M.L. Liu, Distributed computing, principles and applications, Beijing, Tsinghua Press, 2004.

Ontology-based semantic matching in distributed Active data warehouse

Hua Hu, Lidan Ji, Bin Xu, Chenxiang Yuan
College of Computer and Information Engineering, Zhejiang Gongshang University
Hangzhou, Zhejiang, China
Email:jilidan@gmail.com
Email:huhua@mail.zjgsu.edu.cn

ABSTRACT

Along with the development and population of network, data sources' heterogeneities gradually become a crucial problem. So the need of resolving these heterogeneities is becoming more important, especially semantic heterogeneities. Two data sources often have the different words to describe the same meaning or the same word to present the different meanings. Despite the critical importance, current approaches to semantic matching of heterogeneous databases have not been sufficiently effective. This paper uses ontology technology to get the semantic matching in distributed active data warehouse. Our approach aims at creating ontologies and measuring semantic similarities among concepts presented in different ontologies. Then a process of whole similarity matching is presented and validated. From this method, the precision which describes how much semantic heterogeneity can be resolved between two data sources is higher than other way.

Key words: Ontology, Semantic Matching, Distributed Active Data Warehouse, Heterogeneous Data Sources.

1. INTRODUCTION

The rapid development of computer network and communication technology has produced many data warehouses with multiple distribute data sources. Although it has been researched and many wonderful results have been achieved in these years, the database integration is still a difficult task [1,2] due to the various heterogeneities among sub-databases and data warehouses [3]. The crucial problem now is how to resolve heterogeneities automatically in data warehouses, especially in distributed active data warehouses, which need to draw the data automatically, and no people participate to resolve these heterogeneities.

Typically, there are two types of heterogeneities in database inter-queries. One is structure heterogeneity and the other is semantic heterogeneity. Structure heterogeneity refers to the differences among local definitions, such as attribute types, formats and precisions. To be mentioned, some data structure in the sub-databases may not be the same as that in the data warehouse. These differences can be easily resolved [4]. Semantic heterogeneity refers to the differences or similarities in the meaning of the data among the sub-databases and the data warehouses. For example, two concepts in different data sources could have the same meaning, but with different names. Therefore, the system should realize that the two concepts are actually the same one when querying. Alternatively, two concepts in two data sources might be named identically, while their meanings are incompatible. Hence, these concepts should be differentiated during querying or integration.

This paper puts forward a method to resolve the problems arisen above. We create ontologies and make use of functions, and implement semantic matching by a process of ontology matching.

The remainder of this paper is organized as follows. In Section 2, we present a simple commercial database example of semantic mismatching. Section 3 provides a simple definition of ontology and its implementation. In Section 4, we outline an ontology-based semantic matching in database warehouse. We resolve the problem of semantic mismatching and discuss approach to measure and integrate similarity. Section 5 describes the process of semantic matching and gets the results. The conclusion is stated in Section 6.

2. SEMANTIC MISMATCHING IN COMMERCIAL DATABASE

In this section, we present a simple commercial database example of semantic mismatching. There are a great of sub-companies, which maintain stocks of various products [5]. Two different companies (including sub-company and the central company) of this kind may maintain part and similar information such as those shown in table 1 and table 2.
When we examine these two database tables, we discover several things. First, the fields *StoreID* and *Store NO* refer to the same thing. Second, the fields *Product Item* and *Product Name* probably refer to the same concept. Looking at the fields, we really couldn't determine whether these fields use the same unit. Take a point instance, in the Table 1 field *Total Money* may be in US dollars, while in Table 2 may be in *RMB*. In addition, *shirt* may be a part of *suit*. *Trousers* and *pants* probably mean the same things. And *Tom* probably is *Tom Smith*.

Users who want to gather the data that spanning these two databases through data warehouse would probably like answers that have resolved the above types of semantic ambiguities. For example:
If a user wants to find all products that named *Skirt* from both table 1 and table 2, the system should be smart enough to know that the field names *Product Name* and *Product Item* are the same.

Likewise, if the user wants to get total money of all suits in the two tables, the system should know that t*rousers* are a part of *suits*. In addition, if the table 1 uses *USD* and table 2 uses RMB to express units, the system should automatically convert the value in *USD* to *RMB*, vice versa.

In this paper, we address these issues and a function to get semantic matching successfully.

Table 1. Sellbill1

Store ID	Customer Name	Product Name	Total Money	Time	……
100	Tom	Suit	2000.23	2005.7.2	……
101	Amy	Coat	1255.25	2005.8.7	……
102	Jim	trousers	1548.26	2005.8.2	……
103	Tom	Skirt	1654.21	2005.8.9	……
104	Kate	Sports Shirt	4852.29	2005.9.2	……
……	……	……	……	……	……

Table 2. Sellbill2

Store NO.	Customer NO	Product Item	Total Money	Date	… …
00100	Tom Smith	T Shirt	2000.4	2005-7-21	… …
00101	Amy	Skirt	1455.2	2005-8-7	… …
00105	Jim	trousers	1549.1	2005-8-25	… …
00103	Tomas	pant	1654.9	2005-8-28	… …
00108	Kate	sweat	4853.3	2005-9-26	… …
……	……	……	……	……	… …

3. ONTOLOGY

3.1 Definition

Ontology is defined as “specifications of a shared conceptualization of a particular domain” [10]. It is mainly used to help the designers to understand the semantics of database objects. Ontology specifies a conceptualization of a domain in terms of concepts, attributes, and relations. People and application systems could communicate by them. Thus they facilitate knowledge sharing and reusing [11]. In our model，ontology means a taxonomic and hierarchical vocabulary of some fields. Typically each node represents a concept and each concept is a specialization of its parent in the taxonomy tree. A simple ontology as fig. 1.Here the data is organized into taxonomy that includes Store ID, Customer Name, Product Name, Total Money. Skirt has attributes such as size, color and style.

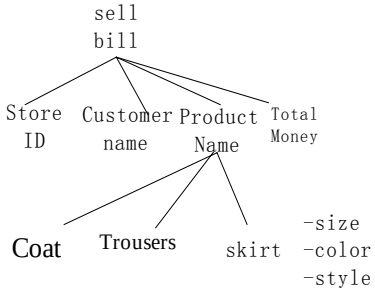


Fig.1. Ontology of sellbill1

3.2 Creating ontologies

There are three metamodels that represent more structural or descriptive representations those are somewhat less expressive in nature than CL（Common Logic） and some DLs （Description Logics）. These include metamodels of the abstract syntax for RDFS [RDF Schema], OWL [OWL Reference; OWL S&AS], and TM (Topic Maps, [TMDM]). RDFS, OWL and TM are commonly used in the semantic web community for describing vocabularies, ontologies and topics, respectively [15].

We use the software protégé 2000 to create our ontologies. From the schema of database and the corresponding model of ER, we could easily get the data source’s terms and relationship between them. In the table 1 and table 2, there are relative terms: Store, Customer, Product, Money, and Time. The relation between them: Customer has products; every customer belongs to different stores; every product has his price; plus the entire price is the total money; every customer and his products decide a bill and every bill will have a making time. In addition, every term has its own attributes. For example, customer has name, age, address, phone number and credit. After picking up all these terms, attributes and relations, we can create ontology.

The ontology’s creating strategies base on the schema of database is:

(1) Every table is a class, and the class name is equal to the table name.

(2) Table’s fields are classes’ attributes, and their attributes’ names are equal to fields’ names.

(3) If there is a field that has foreign key, the field will be defined as the class’s attribute and his type is objecttype.

(4) We also need to define the values of domain and range for appointing the two tables that have the foreign key relation. All of the other fields’ types are datatype.

From the above strategies, the framework of ontology is as fig.2.

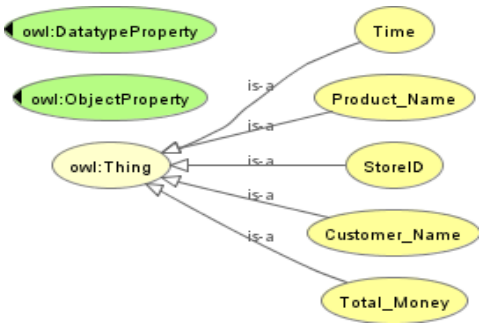


Fig. 2. Table sellbill1’s ontology picture

Some parts of the Ontology's owl file are as below:

```
<owl: Ontology rdf: about=""/>
 <owl: Class rdf: ID="Customer_Name"/>
 <owl: Class rdf: ID="StoreID"/>
 <owl: Class rdf: ID="Product_Name"/>
 <owl: Class rdf: ID="Total_Money"/>
 <owl: Class rdf: ID="Time"/>
 <owl: ObjectProperty rdf: ID="is_a">
  <rdfs: domain rdf: resource="#StoreID"/>
                <rdf: type
   rdf:resource="http://www.w3.org/2002/07/owl#T
                ransitiveProperty"/>
              </owl: ObjectProperty>
  <owl: DatatypeProperty rdf: ID="datatype">
   <rdf: type
rdf:resource="http://www.w3.org/2002/07/owl#Ann
```

4. ONTOLOGY-BASED SEMANTIC MATCHING

4.1 Definition

A definition of similarity from Merriam Webster's Dictionary is: having Characteristics in common: strictly comparable. We want to strictly compare two entities to find identity among them. Intuitively, two entities need common characteristics to be similar. We also get a formal definition of similarity here derived from [18]:

Sim (x, y) ∈ [0, 1].
Sim (x, y) =1, iff x=y.
Sim (x, y) =0, if two objects are different and have no common characteristics.
Sim(x, y) = Sim(y, x): similarity is symmetric.

4.2 Similarity methods

1) Labels

The first level describes entities is "NO. Label" that are human identifiers (names) for entities, normally shared by a community of humans speaking a common language. We can therefore infer that if labels are the same, the entities are probably also the same (see the content in the rectangle below).

```
<owl: Class rdf: storeID="100">
<rdfs: label>store number</label>
</owl: Class> <owl: Class rdf: ID="00100">
<rdfs: label> store number </label>
```

Even without being able to interpret "label", it is still possible to do comparisons based on string or number matching.

2) Similarity measure

In general, a matching may be specified as a query that transforms instances from ontology into another [9]. There are many ways to formulate a matching problem for taxonomies. The specified problem that we consider is as follows: given two taxonomies and their associated data instances, for each node (i.e., concept) in taxonomy, and the most similar node in the other taxonomy, for a similarity measure.

In this measure, the notion of the joint probability distribution between any two concepts A and B is well defined. This distribution consists of four probabilities: P (A, B); P (A, $\overline{B}$); P ($\overline{A}$, B), and P ($\overline{A}$, $\overline{B}$). A term such as P (A, $\overline{B}$) is the probability that a randomly chosen instance from the universe belongs to A but not to B, and is computed as the fraction of the universe that belongs to A but not to B. Many practical similarity measures can be defined based on the joint distribution of the concepts involved. For instance, a possible definition for the "exact" similarity measure is Eq. (1) as follow:

$$Jaccard - sim(A,B) = P(A \cap B)/P(A \cup B) = \frac{p(A,B)}{P(A,B)+P(A,\overline{B})+P(\overline{A},B)} \quad (1)$$

This similarity measure is known as the Jaccard coefficient [16]. It takes the lowest value 0 when A and B are disjoint, and the highest value 1 when A and B are the same concept. Most of our experiments will use this similarity measure [6].

If computer the similarity by the formula above, we need to know P (A, B), P (A, $\overline{B}$), P ($\overline{A}$, B), P ($\overline{A}$, $\overline{B}$). P (A, B) is the ratio of the instances that belong to A and B, and the formula is Eq. (2) as below. P (A, $\overline{B}$), P ($\overline{A}$, B), P ($\overline{A}$, $\overline{B}$) can be calculated by the same way. We use U1 to describe all instance of O1. N (U1) means the number of instances. N (U1AB) denotes the instances belong to A and B in U1.

$$p(A,B) = \frac{N(U_1^{AB}) + N(U_2^{AB})}{N(U_1) + N(U_2)} \quad (2)$$

After calculate these ratios we can use the formula of Jaccard to computer the similarity. Input ontology O_1, $O_2$and their instances, using the method above, the total ratios that have to computer is 4| O_1 | | O_2 |, there into | O_1 | | O_2| denote the number of $O_1$and O_2 respectively [7].

4.3 Integration approach

With the single similarity methods some more steps are required to receive the best possible matching. In this paper we use the method of cut-off to integration.

As described earlier, goal of this approach is to find correct matching between two ontologies. With the above described we have a list which consists of the most similar entities of two ontologies and the corresponding similarity value. Now we should decide which level of similarity is appropriate to indicate equality for the matching and which strongly indicates inequality? It is important to make the decision where to put the cut-off. Every similarity value above the cut-off indicates a match; everything below the cut-off is dismissed.

We utilize the method of Delta to get the value of cut-off Eq. (3). In this method the cut-off value for similarity is defined by taking the highest similarity value of all and subtracting a fixed value from it. Again this can be found through maximizing over different measures.

$$Cut\ -off\ = \max\left(sim\left(ex, ev\right)\middle|\forall\, ex \in \mathrm{O}1, ev \in \mathrm{O}2\right) - c \quad (3)$$

5. IMPLEMENTATION

5.1 Process

So far we have presented several ideas of how to create ontologies and compare two entities. Then match between them. Now we use the following fig. 3 to describe the entire process of ontology matching. The steps are as follow:

1) Input two ontologies, which have to be matched .We, will therefore calculate the similarities between any valid pair of entities.

2) Calculate the similarities via measures, which are independent of other similarities. In our case we adopt the label similarity. If the number labels are the same. Then straightly step to 4. If not, turn to 3).

3) Calculate all entities' values of P (A, B), P (A, $\overline{B}$), P ($\overline{A}$, B), P ($\overline{A}$, $\overline{B}$) , Jaccard-sim (A, B).

4) Integrate the values in step 2 or 3, and calculate the value of cut-off.

5) Compare the value of Jaccard-sim (A, B) and cut-off. If Jaccard-sim (A, B)>cut-off, then matching, else mismatching.

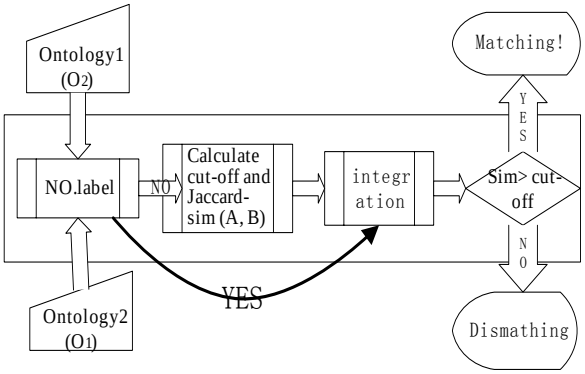


Fig. 3 The process of ontology matching

5.2 Result

From all the presented methods and actions one can determine the similarity between two entities and decide if they can be regarded as equal or not. These will then be tested. The implementation itself was done in Java using the ECLIPSE frameworks for ontology access and maintenance. All the tests were run on a standard Pentium 4 2.4GHz computer.

Besides our own ontologies "sellbill", the others are from http//www.daml.org/ontologies/. The definitions of precision and recall are adapted by us to measure the experiment result [17]. Recall describes the number of correct matching found in comparison to the total number of existing mappings as Eq. (4).

$$r = \frac{\#\, correct\ \_\ found\ \_\ mapping}{\#\, possible\ \_\ existing\ \_\ mapping} \quad (4)$$

Precision we measure the number of correct mappings found versus the total number of retrieved mappings (correct and wrong) as Eq. (5).

$$p = \frac{\#\, correct\ \_\ found\ \_\ mapping}{\#\, possible\ \_\ existing\ \_\ mapping} \quad (5)$$

The result of the experiment is as Table 3:

Table 3. Result of the experiment

Ontology Name	Experiment result	
	Recall	Precision
sellbill	0.705	0.732
standford	0.648	0.725
man	0.653	0.736
cyc	0.732	0.714

6. CONCLUSION

This paper describes the definition and the designing of ontology. The main objective is to improve the capabilities of semantic matching by measuring similarities. It contributes to the information integration and knowledge acquisition from heterogeneous, distributed information sources. We have shown a methodology for identifying matching between two ontologies by several similarity measures.

7. ACKNOWLEDGEMENT

This research was completely supported by Science and Technology Department of Zhejiang Province with No. 2005C21025 to Dr. Hua Hu. Thanks to our team for the fruitful discussions and efficient work. Thanks to the people who have cooperated with us in this paper including Haobin Zhang and Shunjian Ni from Zhejiang Gongshang University.

8. REFERENCE

[1] C. Batini, M. Lenzerini, et al, A comparative analysis of methodologies for database schema integration, ACM Computing Surveys, 1986, 18(4): 323-364.

[2] A. Sheth, A..Larson, Federated database systems for managing distributed, heterogeneous and autonomous database, ACM Computing Surveys, 1990, 22(3): 183-236.

[3] Z.B. Liu, X.Y. Du, et al, Integrating Databases, Internet 1998 Second International Conference on Knowledge-Based Intelligent Elecnonic System, April 1998, No.21-23: 381-385.

[4] Farshad Hakimpour, Andreas Geppert, and Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach.

[5] P. Bonatti, Y. Deng, et al, An Ontology-Extended Relational Algebra, IEEE.2003.

[6] A. Doan, J. Madhavan, et al, Learning to Map between Ontologies on the Semantic Web.

[7] Y. Cheng, H. Huang, et al, Similarity measure-based

dynamic concept mapping algorithm.
[9] Christiane Fellbaum, WordNet: An Electronic Lexical Database.
[10] Gruber. T. R., Toward Principles for the Design of Ontologies Used for Knowledge Sharing, International Journal of Human and Computer Studies, 43(5-6): 907-928.
[11] L. Zhang, J.G. Gu, Ontology Based Semantic Mapping Architecture, Proceeding of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005, 2200-2202.
[12] Marc Ehrig, York Sure, Ontology Mapping - An Integrated Approach, 6-7, 11-12.
[13] Q. Lei, Ontology-based Heterogeneous Data Integration Research, Master paper, 27-29.
[15] Ontology Definition Metamodel, Third Revised Submission to OMG/RFP ad/2003-03-40 Submitted by: IBM Sandpiper Software, Inc, 53-54.
[16] Van Rijsbergen, Information Retrieval. London: Butterworths, Second Edition, 1979.
[17] H. Do, S. Melnik, et al, Comparison of schema matching evaluations, Proceedings of the second int. workshop on Web Databases (German Informatics Society), 2002.
[18] G. Bisson, Why and how to define a similarity measure for object based representation systems, Towards Very Large Knowledge Bases, 1995, 236–246.
[19] H. Takeda, K. Iwata, et al, An Ontology-Based Cooperative Environment for Real World Agents, Proc. Int'l Conf. Multi agent Systems, Dec. 1996, 353-360.

Running Existing Applications on Grid*

Guo Qingping1, Raihan Ur Rasool1,2
qpguo@mail.whut.edu.cn, vitalfact@hotmail.com
1Wuhan University of Technology, Wuhan, 430063, P.R.China
2NUST Institute of IT, Pakistan

ABSTRACT

Grid computing is perceived as having as much potential for changing the way companies do business as the Internet did. But it presents some new challenges. A normal user spends too much time on procedures for accessing a resource and too little time on using the resource itself. In this paper we have presented a Grid-Operating System; 'Users-Grid', being developed at our lab, lets existing applications to run on the Grid and seamless shift of computational load from user's computer to other nodes on the Grid. It provides computing power transparently and easily by true virtualization. A working prototype of one part of this project is ready and has been demonstrated here.

Keyword: Grid Computing, Distributed Computing, Agents, Service Oriented Architecture and Grid-OS.

1. INTRODUCTION

Internet is a network of communication, whereas grid computing is seen as a network of computation. Grid technology allows organizations to utilize numerous computers to solve problems by sharing computing resources. However it provides only fundamental services; and lacks sufficient abstraction at the application level. Therefore, we cannot say that we are using a true virtualized infrastructure, which is the driving force behind grid computing. Abstraction of various Grid capabilities is needed and existing applications need to be able to take advantage of Grid infrastructure transparently.

Even if the computer is connected to the Grid, it is difficult for the research professionals and normal users who are not computer experts to study the mechanics of the Grid and then to use it in a special way, which adds further complexity in their work.

Our motive was to 'put the technology in the pocket'; so that every one could get benefit of Grid not only research labs. We proposed our unique idea 'Users-Grid' [1], an Open source Grid-Operating System. Based on Globus and Condor, using.

Service-Oriented architecture and Agent Technology, it gives the feature of true virtualization to users. Users-Grid enables automatic and seamless shift of extra load of computation from user's computer to the other nodes on Grid, which are underutilized and bill back the user according to the resource usage.

In this way user will utilize Grid resources transparently without concerning how and where the job is submitted. This virtualized infrastructure allows applications to exploit heterogeneous resources across the enterprise.

* This work has been supported by the Natural Science Foundation of HuBei Province (NSFHB 2005ABA 227)
Corresponding author: Guo Qingping

An application is said to be grid-enabled when it can simply run in a grid. Users-Grid provides seamless and transparent access of Grid resources to existing applications like Microsoft Excel, Mat lab, etc. It provides rich APIs support, which is utilized by agent to analyze the 'application flow', Job types and tasks within an application.

In this paper we have not explained the architectural details rather we have focused on 'how it works', architectural details can be found at [1]. The remaining sections of this paper present background and related work, understanding Users-Grid, strategies and working, conclusions and future work.

2. BACKGROUNDS AND RELATED WORK

- Active Sheets, it is a mechanism for parallelizing traditional spreadsheets, it is component-based interface for spreadsheet. It implements a two stage evaluation process for custom functions, however
  - ◊ It works with Nimrod and Netsolve grid middleware systems and provides parallelism at the spreadsheet function level.
- Inner Grid Nitya, It is a commercial product from GridSystems Company, it grid-enables Excel, by distributing spreadsheet calculations among corporate PCs.
- Platform Symphony introduced the Adapter for Microsoft Excel, which aims to, accelerates online and batch operations for business critical financial services.
  - ◊ Platform Symphony is specifically designed for Enterprises, to pre-allocate number of clusters.

However,

- o Three above-described systems are not open-source; they do not provide rich API support for other existing applications, because these are neither a RMS nor complete Grid-Operating Systems.
- o No feature of automatic shifting of extra load of computation from user's computer to the other nodes on Grid and billing back to user seamlessly.
- o These projects are mainly focused on enterprise level instead of taking in mind the global distribution of load across the Grid.
- o These commercial softwares mainly only focus on MS Excel to enable it to run on Grid, without concern to other many important existing applications.

A brief listing of other concerning software systems is provided here.

- The Sun Grid Engine (SGE) [3] is based on the software developed by Genias known as codine/GRM. A user submits a job to the SGE, and declares a requirement profile for the job. When the queue is ready for the new job, the SGE determines suitable jobs for that queue and then dispatches the job with the highest priority or longest waiting time; it will try to start new jobs on the most

suitable or least loaded queue. It has open source as well as commercial versions. It is also a RMS.

- ◊ The software has been ported to many operating systems, including Solaris, Linux, IRIX, Tru64, AIX, HP/ux, but it is not supported (open source SGE) on Windows [4].
- ◊ It is batch processing only, you don't get rich-API support which makes easy to integrate it into your existing application environment.
- ◊ It also has no facility of automatic load shifting and Plug-in to make existing applications to run on Grid. Hence it is not a complete solution.

PBS [5] and LSF also fall in this category, these are commercial products. Maui and CSF are also in the list of these successful and mature schedulers, however these batch processors and schedulers share the common limitations with SGE as described above.

- Vega PG [6], its main idea is to construct the Grid as a virtual computer in wide area. The Vega PG adopts a lightweight architecture, which uses simplified structures and protocols. Vega maps between the Grid and traditional computer systems and use the virtual computer concept to build the Vega PG on legacy systems. It borrows many mature technologies from the traditional computer architecture design. However,
  - ◊ It has also no facility to enable existing applications to run on Grid without code change, hence no plug-in support to applications is provided.
  - ◊ It has no feature of automatic shifting of extra load of computation from user's computer to the other nodes on Grid and billing back to user seamlessly.

Non of the above described systems has the feature of automatic shifting of extra load of computation from user's computer to the other nodes on Grid, which are underutilized and bill back the user according to the resource usage. Moreover none of the above-described systems allows you to take your existing business critical applications (not only MS Excel) and run them on the grid without minor changing the code. On the other hand Users-Grid is an easy to use and totally transparent complete system, which is not just a toolkit or RMS. It contains a complete accounting and billing system based on resource usage.

3. UNDERSTANDING USERS-GRID

If the user has Users-Grid installed on his computer and he is already connected to Grid then whenever he would be in lack of computational or data resources, Users-Grid can seamlessly transfer extra burden of computation on other computers on Grid. Users-Grid provides plug-in support; especially for Excel and Matlab users to transfer bulky spreadsheets computation on other computer and having results back. And depending on administrator's policies the user will be charged bill according to his resource usage. The user can also use Users-Grid GUI facilities to submit the jobs on the Grid and see the progress. The Users-Grid resource management system would automatically take care of which resource is suitable to run the job. It will automatically make scheduling policies.

Users-Grid uses Agent Technology to know about extra computational burden, or free resources on any computer onto Grid. Agents cooperate with each other to balance workload in the global grid environment using service advertisement and discovery mechanisms. Both local schedulers and agents can contribute to overall grid load balancing, which can significantly improve grid. The architecture of Users-Grid is presented in fig. 1.

Users-Grid actually uses three already developed and mature technologies from the grid community. It uses Globus Toolkit and Condor at its Base Layer. In addition to these two already developed systems, it uses an accounting system that tracks resource usage. We chose these three systems on the basis of maturity, being open source and having implementation on many known operating systems. The details about its Architectural components can be found at [1]

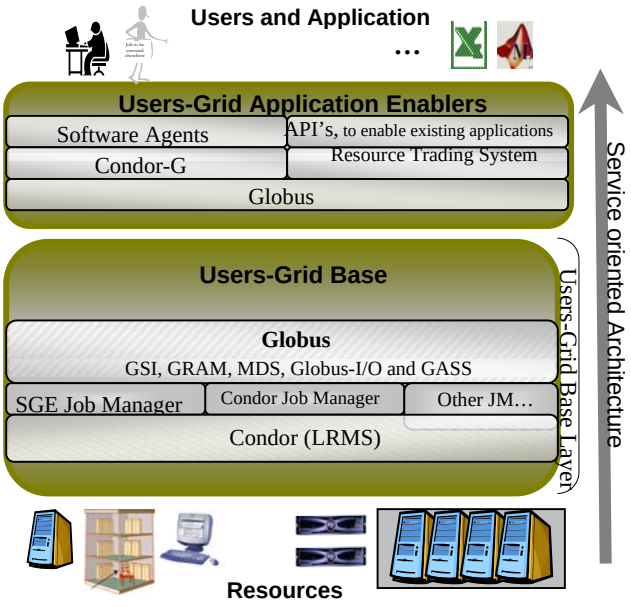


Fig. 1. Users-Grid Layered Architecture

Condor is a specialized job and resource management system (RMS) for compute intensive jobs. The combination of the Globus and Condor software packages presents an attractive solution for Grid applications. Globus includes the ability to use Condor as a back-end scheduler. Users may submit jobs to a single Globus resource, which is actually a Condor cluster. Then, Condor sends that job to whichever machine is available. Meanwhile, the user on the submitting end can remain blissfully unaware that Condor is being used [2].

Any other local resource management (LRMS) software can be used, however we choose Condor because of its wide range implementations in academia and industry. The remote host is not required to run any Condor software; rather, jobs are submitted to the remote host's Globus job manager in the usual way. The combination of Condor-G, Globus, and Condor results in the ease of use and fault tolerance of Condor; combined with the security, authentication, and encapsulation of Globus [2]. 'Gold' is an open source accounting system that tracks resource usage on High Performance Computers. It provides balance and usage feedback to users, managers, and system administrators [7]. Considering its features and characteristics we found it best to be used with Users-Grid after few changes in the code.

4. STRATEGIES AND WORKING

Strategies

Application Enablers' layer is actually engine of Users-Grid which enables existing applications to run on the grid. Some like to think of a grid as a distributed cluster. Although such parallel applications certainly can take advantage of a grid, nobody should dismiss the use of grids for other types of applications as well. Even a single threaded batch job could benefit from a grid environment by being able to run on any of a set of systems in the grid, taking advantage of unused cycles. The need for large amounts of data storage can also benefit from a grid. Examples include thoroughly analyzing credit data for fraud detection, bankruptcy warning mechanisms, or usage patterns of credit cards.

To take advantage of multiple resources concurrently in a grid, we have to consider whether the processing of the data can happen in parallel tasks or whether it must be serialized and the consequences of one job waiting for input data from another job. At this stage it is also very important to know about Application flow and Job flow. Application flow is the flow of work between the jobs that make up the grid application. The internal flow of work within a job itself is called job flow. So analyzing the type of flow within an application delivers the first determining factor of suitability for a grid. Besides the flow types, the sum of all qualifying factors allows for a good evaluation of how to enable an application for a grid. There are three basic types of application flow that can be identified: which are Parallel, Serial and Networked [8]. When dealing with mathematical calculations the commutative and associative laws can be exploited. In iterative scenarios (for example, convergent approximation calculations) where the output of one job is required as input to the next job of the same kind, a serial job flow is required to reach the desired result.

For a given problem or application it would be necessary to break it down into independent units. To take advantage of parallel execution in a grid, it is important to analyze tasks within an application to determine whether they can be broken down into individual and atomic units of work that can be run as individual jobs [8]. As grid standards such as OGSA mature, grid enablement will mean that the application can run as a Web service in a grid environment, while optionally taking advantage of the various services provided by the grid infrastructure. For example, Java 2 Platform, Enterprise Edition (J2EE) or C application run as a Web service on top of grid-enabled middleware, such as the upcoming version of 'WebSphere Application Server', while optionally taking advantage of the services provided by the middleware and the grid infrastructure [9].

We used few strategies for grid enablement. For instance, an existing application can be enabled in a way that its most resource-intensive computational pieces run as a Batch Anywhere job on any suitable computer in the grid. The 'Users-Grid Base' decides which machine to assign at run time. Then, by eliminating concurrence inhibitors, the same application can evolve so that its grid-enabled piece runs as an Independent Concurrent Batch instance of the same batch job. It supports multiple independent instances of the same application running concurrent. This gives the overall application far more throughput than having only one instance running at a time. It could eliminate more concurrence inhibitors and allow its grid-enabled piece to run as a Parallel Batch job. Notice that what might be changing is the definition of a work unit, not the application. A work unit in this case would be an array of the original work units used to implement the first two strategies for enablement [9]. Two components must be added: one subdivides the work and hands it to the grid. The other aggregates results. In Users-Grid Agent aggregates the result with the help of APIs.

Another possibility is to leave the batch world behind but this strategy requires change in the code of existing application. Agent uses Users-Grid's APIs to enable many existing applications to get benefit from Grid.

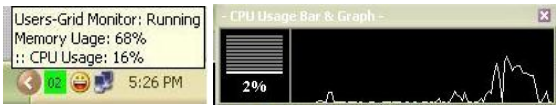


Fig. 2. Users-Grid agents monitoring the resource usage

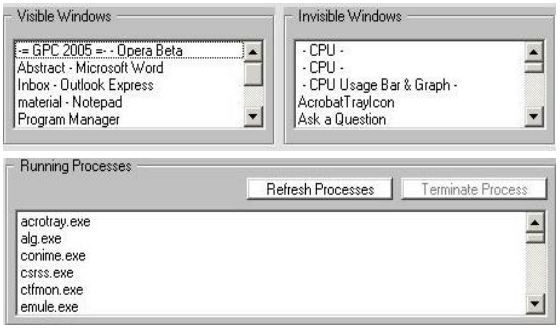


Fig. 3. Agent takes care of all the applications and processes running on user's system

Users-Grid: how it works

In Users-Grid agent resides like a daemon in user's computer and monitors the resource usage (fig. 2.) continuously and whenever resource usage of user's computer is more then pre-set limit (for a specific period of time, according to policies by user), it does the following steps with the help of Users-Grid's APIs,

1. Check the types of applications currently running and analyze the type of flow within an application.
2. Analyze the task within an application.
3. Is there any one, which can be enabled to run on Gird? If yes, then
4. Analyze Job criteria (Batch Job, Interactive Job, Standard Application, Parallel Application)
   a. Either, analyze to determine how best to split the application into individual jobs, maximizing parallelism and then break the application's computation into independent, individual and atomic units of work that can be run as individual jobs, and submit to 'Job submit Module' of 'Users-Grid Application Enablers layer'.
   b. Or, shift the whole job (instead of splitting it) to remote node, which has the facility to run this job, with the help of 'Job submit Module' of 'Users-Grid Application Enablers layer'.

Agent takes care of the all the applications (visible or on background) and running processes (fig. 3.). Users-Grid's architecture includes Condor and Globus; it includes separate Server and Client installations packages. 'Users-

Grid client' can be configured as both 'submit' and 'execute' machine. 'Application Enabler Layer' is very important to be included; however 'Users- Grid Base layer' has not been included in Client installation package. For server installation, many optional components of 'Application Enabler Layer' may not be installed. Users-Grid can submit the jobs to other nodes of the grid even if those nodes do not have Users-Grid installed. But these nodes should have Globus and local Resource Management Software installed already

User can make policy when the Users-Gird should activate (fig. 4.). So whenever the resource (CPU or memory) usage would be more then a certain limit, Users-Grid agent would start analyzing if any application currently running can be enabled for the Grid, according to above described rules. Users-Grid creates a log file to know when actually the load shifting had started (fig. 5.).

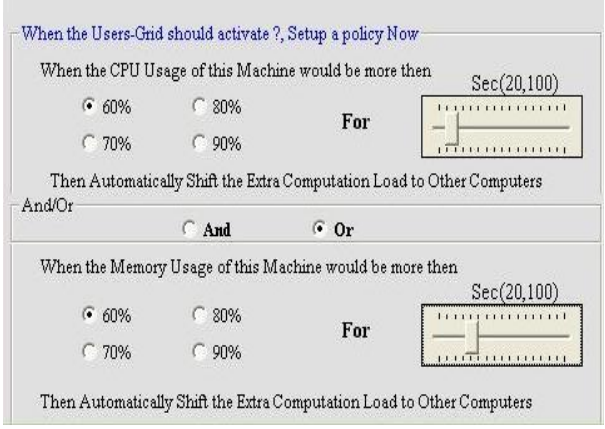


Fig. 4. Users-Grid policies Form

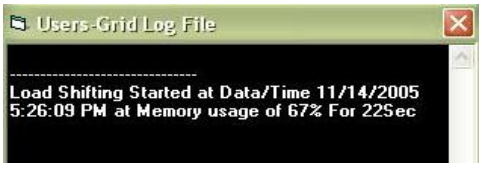


Fig. 5. Users-Grid Load Shifting Log

Excel in Focus: While Excel provides a powerful and flexible environment for building formula and macro-based applications, the performance of these applications are limited by the fact that Excel runs in a desktop personal computer environment. To accelerate performance of these applications, Users-Grid provides the ability to run Excel spreadsheets in parallel on a grid environment. Users-Grid allows Excel applications to run on a grid with little or no modification. This plug-in support has been provided on Windows platform only.

5. CONCLUSION

The technologies, softwares or toolkits available today to realize the idea of Grid, make it more difficult to use for normal user. We need to build supporting environment around these toolkits to get a better level of abstraction and to simplify its use. We have to enable existing business and science applications, because we cannot just start application development from the scratch.

We have presented 'Users-Grid', a Grid–OS being developed at our lab; and described its complete working; it runs existing applications on Grid, it provides Plug-in support to many existing applications to make them grid-enabled. User needs not to find the available resource manually and then to submit a job in a special format (RSL). With Users-Grid user only launches the heavy simulation or job on his own computer without knowing the details and complexities of Grid, and Users-Grid will seamlessly submit the job onto vacant machines on Grid. So user's machine would never go beyond the resource usage limit described in policy forms (Fig. 4.). Thus Users-Grid provides true virtualized infrastructure for Grid Computing.

6. REFERENCES

[1] U.R. Raihan, Q.P.Guo, " A Proposal of Next Generation Grid-Operating System", Grid Computing and Applications (GCA'05) Las Vegas USA, 2005.

[2] Globus-Only topology: http://grid.phys.uvic.ca/docs/uberdoc/node11.html (node11.html to node14.html)

[3] Sun Grid Engine: http://www.sun.com/software/Gridware/.

[4] SuSe Linux to Distribute Sun's Grid Engine Software. http://siliconvalley.internet.com/news/article.php/1013251

[5] Open PBS *(it has link to download but yet un-downloadable)* http://www.openpbs.org/download.html

[6] W. Li, Z.W. Xu et al, "The Vega Personal Grid: A Lightweight Grid Architecture", Fourteenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2002)

[7] Gold Accounting and Allocation Manager: http://www.emsl.pnl.gov/docs/mscf/gold/

[8] J. Bart, F. Luis, "IBM-Enabling Applications for Grid Computing With Globus", IBM, 2003, 45-53.

[9] Excel for Business Intelligence, September 9, 2003. http://www.microsoft.com/office/previous/xp/business/intelligence/excel.asp

Parallelizing a Two-Pass Improved Encoding Scheme for Fractal Image Compression

Ong Ghim Hwee，Ching Kin Wah Eugene
School of Computing, National University of Singapore
Republic of Singapore
Email: onggh@comp.nus.edu.sg, chingkin@comp.nus.edu.sg

ABSTRACT

An improvement scheme, so named the Two-Pass Improved Encoding Scheme (TIES), for the application to image compression through the extension of the existing concept of Fractal Image Compression (FIC), which capitalizes on the self-similarity within a given image to be compressed, is proposed in this paper. We first briefly explore the existing image compression technology based on FIC, before proceeding to establish the concept behind the TIES algorithm. We then devise an effective encoding and decoding algorithm for the implementation of TIES through the consideration of the domain pool of an image, domain block transformation, scaling and intensity variation, range block approximation using linear combinations, and finally the use of an arithmetic compression algorithm to store the final data as close to source entropy as possible. Also, we compare the performance of this implementation of the TIES algorithm against that of FIC under the same conditions. Additionally, due to the long encoding time required by the TIES algorithm, this paper then proceeds to propose a parallelized version of the TIES algorithm and its implementation, before finally concluding with an empirical analysis of the speedup and scalability of the parallelized TIES algorithm.

Keywords: Fractal Image Compression, Parallel Processing, Peak Signal-to-Noise Ratio, Linear Combination.

1. INTRODUCTION

1.1 Fractal Image Compression (FIC)

Fractal image compression is a lossy compression technique proposed in the 1980s by M. Barnsley [1, 2, 3]. The technique is based on the observation that since fractals can generate relatively realistic images, then, it should be possible to store a given image in the form of just a few basic fractal patterns, coupled with the specification of how to restore the image using those fractals. The FIC algorithm starts from the complete image, and breaks down that image into a number of partitions. For each given partition P_i, the algorithm then searches the image for other sub-sections of the image, S_j, which are relatively similar to P_i, and then maps P_i to S_j, until a mapping for every sub-section of the image has been found. The compressed image would then consist of all the partitions P_i, and their corresponding mappings. In FIC terminology, P_i is called a domain block, and the corresponding set of partitions is called the domain pool, while S_j is called a range block, and the corresponding set of sub-sections is called the range pool.

1.2 An Improvement Scheme to FIC

This paper proposes an improvement scheme, which is based in part on the earlier described FIC algorithm, which will reduce the overall size of a compressed image as compared to that of FIC under similar conditions [3]. This scheme will, in fact, make extensive of use the FIC algorithm for partitioning a given image into range blocks (sub-images of the original image) and extracting the domain pool by finding a suitable mapping to approximate the range blocks from the domain blocks. However, we note that FIC searches for the best match between a range block and a domain block, and then proceeds to store that mapping of the range block to the domain block. In other words, FIC produces a one-to-one mapping for every range block from a set of domain blocks. This has two consequences. Firstly, the domain to range mapping is the best for a one-to-one mapping, but would it be possible that a linear combination of domain blocks might result in a better approximation of that range block? If so, then image quality can be increased without increasing the size of the domain pool. In addition, by using linear combinations of domain blocks to approximate the range blocks, it also becomes possible to reduce the size of the domain pool should some of the domain blocks now be made redundant. This is similar to the idea of using a resultant vector to represent a linear combination of vectors in the technique of matching pursuit [4]. Secondly, can the size of the domain pool be reduced if we search for similar domain blocks within the domain pool itself?

Indeed, both of these are possible, and this paper proposes an extension and improvement to the original FIC algorithm by capitalizing on these two observations to improve compression performance. However, it is worthwhile to note that while the idea behind searching for linear combinations of domain blocks instead of simply using one-to-one mappings of domain blocks to range blocks seems simple, determining what is considered the best linear combination is extremely computationally expensive, as the algorithm would have to search through all possible linear combinations of varying sizes. Thus, this paper will also propose a simple and effective algorithm to searching for such linear combinations.

2. THE PROPOSED ALGORITHM

2.1 Introduction

The proposed algorithm is a two-pass scheme, whereby the first-pass involves extracting the domain pool (as in the original FIC algorithm), and the second-pass involves utilizing that extracted domain pool to achieve the maximum compression. Due to its two-pass nature, the proposed algorithm is thus named as a "Two-Pass Improvement Encoding Scheme" (TIES). The first-pass of the TIES algorithm is straightforward as it derives directly from the FIC algorithm. In the second-pass, the TIES algorithm performs:

1) Compression of the Domain Pool
2) Partitioning of the image
3) Searching for the best linear combinations
4) Storage of the results

In the presentation of the TIES algorithm in this paper, a square grayscale image is assumed in order to provide a simple framework for the description and implementation of the algorithm. The algorithm can be extended easily to accommodate non-square image by first dividing the non-square image into two parts, the largest possible square I_{sq1}, and the remaining part of the image I_{r1}, and then applying the algorithm on I_{sq1}. By recursively

applying the above steps of dividing I_r again into I_{sq2} and I_{r2}, it is clear that we will finally be able to apply the compression algorithm on the entire image. In addition, the algorithm can also be easily extended to colour images. Colour images are simply 24-bit images consisting of the layers: red, green and blue (for RGB images). As such, the algorithm can be applied to each layer in succession without modification. Hence, it suffices to describe the algorithm for a square, grayscale image, for clarity.

2.2 Quad-Tree Partitioning

A simple way to partition an image is simply to break the given image up into fixed-sized range blocks, R_i. However, such a method of partitioning has a weakness – there are some parts of an image where there is less detail (for example, a background scene). Hence, larger range blocks will suffice to cover that area well. This will, in turn, reduce the number of domain blocks needed to cover the image as a whole, as well as the number of domain to range block mappings, thus achieving better compression. Likewise, there are also other regions of that same image, which are difficult to cover well using a range block of fixed size. Such regions usually require smaller range blocks in order to capture the finer detail of that portion of the image (for example, the eyes of a person).

Hence, to allow for varying range block sizes, quad-tree partitioning is used in the implementation of the TIES algorithm. In quad-tree partitioning, a square in the image is sub-divided into four equally sized squares when it is not well covered enough by a domain. The measurement of well coveredness is determined by the tolerance factor of the TIES encoder, and is described in Section 3. This process repeats recursively starting from the original image, and continues until a given square (the range block) is small enough to be well-covered by domain block(s). Small squares can be covered better than large ones because contiguous pixels in an image tend to be highly correlated [3].

3. THE TIES ENCODER

3.1 Extracting the Domain Pool

The extraction of the domain pool is similar to the inference algorithm used by the FIC encoder. The original image is first divided into a number of overlapping sub-squares called D_i of size 2^n by 2^n, where $n = 0, 1 \ldots \log_2(\text{size of image})$. The collection $\boldsymbol{D}$ of these sub-squares forms the initial set of domain blocks, or the initial domain pool. Next, the original image is then partitioned, using quad-tree partitioning, into four equally sized sub-sections R_i called range blocks. For each R_i, the algorithm tries to find a best match of D_i from $\boldsymbol{D}$, such that the tolerance criterion is met. If such a match is found, then a mapping is made between D_i and R_i, and D_i is placed into another collection $\boldsymbol{D'}$, which contains the index of the corresponding domain block in $\boldsymbol{D}$ chosen during the extraction process. $\boldsymbol{D'}$ thus is the set of indices representing the final domain pool. If a match cannot be found, then R_i is again sub-divided using quad-tree partitioning into R_{i1}, R_{i2}, R_{i3} and R_{i4}, and for each R_{ij}, where $j = 0, 1, 2, 3$, the algorithm is again applied. This recursive process finally terminates when all R_i have a mapping to a certain D_i, and the collection $\boldsymbol{D}$ is the final domain pool.

3.2 Compressing the Domain Pool

Once the domain pool $\boldsymbol{D}$ has been extracted, we then proceed to compress the domain pool.

The key in this algorithm is to find all domain blocks, which are similar. Two domain blocks, D_a and D_b, are considered to be similar if the sum of the square of the difference in pixel values between corresponding pixels in D_a and D_b is less than a specified tolerance T, for all possible 90^o rotations and flips applied to D_b. This means that the comparison between D_a and D_b has to be applied a total of eight times, four times for each 90^o rotation of D_b without flipping D_b, and another four times with a horizontal flip applied to D_b. Clearly, when making comparisons between D_a and D_b, we also have to ensure that D_a and D_b are of the same size.

3.3 Searching for the Best Linear Combination

One of the issues in finding the best linear combination (LC) is the fact that exhaustively searching through all possible combinations for all possible sizes is computationally expensive, and hence, doing such an exhaustive search would clearly make the encoding process excessively long. Hence, this paper will propose an algorithm that can be implemented simply and efficiently.

The proposed algorithm is implemented by restricting that the reconstruction of the final range block R from its corresponding linear combination of domain blocks $\boldsymbol{L} = \{D_1, D_2 \ldots D_n\}$ is given by taking the sum across all elements of L (i.e., $R = D_1 + D_2 + \ldots + D_n$). On the each iteration, the algorithm then computes,

$$Remainder = R - \sum_{i=0}^{n} D_i \qquad (1)$$

and continues to iterate infinitely, each time choosing another domain block D_{i+1} from $\boldsymbol{D}$ to add to the previously computed sum of domain blocks, but only if the addition of the next domain block will bring the remainder closer to zero. If no such domain block D_{i+1} can be found, the algorithm terminates for this particular range block R. In addition, the algorithm will also terminate when the remainder is less than a tolerance value T_{LC}. Thus, this tolerance value represents whether a given range block is adequately represented by its linear combination of domain blocks.

In addition, when performing the comparison between domain blocks D_{i+1} to choose the best domain block, the algorithm also applies two flip transformations, four rotation transformations, and a number of intensity multipliers on each domain block. Visually, the intensity multiplier simply lightens the image associated with the domain block, such that $D'_{i+1} = I \cdot D_{i+1}$, where I is the intensity multiplier, and D'_{i+1} is the modified domain block.

Finally, since the set of domain blocks $\boldsymbol{D'}$ has been chosen, the original domain pool $\boldsymbol{D}$ can be further reduced in size to contain only those blocks which are elements of $\boldsymbol{D'}$. In other words, we want $\boldsymbol{D} = \boldsymbol{D'}$. Hence, we now perform a second level compression of the original domain pool $\boldsymbol{D}$ to restrict $\boldsymbol{D}$ to elements of $\boldsymbol{D'}$, and we also adjust the pointers in $\boldsymbol{D'}$ such that they point at the correct (changed) element in $\boldsymbol{D}$. This hence reduces the size of $\boldsymbol{D}$.

4. THE TIES DECODER

The implementation of the TIES decoder is fairly straightforward as compared to the encoder, with the only slightly more challenging task being the means to decode the range block location based on the encoding format. Essentially, the decoder reads the encoded file, extracts from the five encoded blocks the matrices and storing them back into the sets $\boldsymbol{D}$, $\boldsymbol{D'}$, $\boldsymbol{T'}$, $\boldsymbol{I'}$ and $\boldsymbol{LC'}$ by implementing an arithmetic decoder, inferring the range block locations and storing them in sets $\boldsymbol{R_x}$ and $\boldsymbol{R_y}$, and finally reassembling the final image using the reconstructed information.

Once the decoder has obtained the final image in terms of raw pixel values, the decoder then performs post-processing and smoothing to remove any blocking artifacts found due to the partitioning of the image into discrete range blocks during the encoding phase, as well as to visually enhance the image. Upon completion of the post-processing, the image is written to an output file in BMP format.

4.1 Decoding the Encoded Data

Obtaining ***D***, ***D'***, ***T'***, ***LC' 'and I*** is performed by using an arithmetic decoder to decode the output file produced by the TIES encoder. However, to construct $\boldsymbol{R_x}$ and $\boldsymbol{R_y}$, we require an algorithm for uniquely inferring the location of each range block from the encoded file.

In general, the corresponding information for each range block is stored in ***LC'***, ***T'***, ***I'*** and ***D'*** in a top to bottom, right to left fashion, regardless of the size of the range block. However, we do know the size of the particular range block since the size is stored in ***LC'*** as well. As such, an example of the decoding of range blocks can be graphically represented in Fig. 4.1.

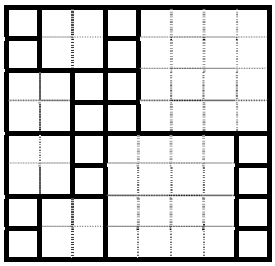


Fig. 4.1. Example of range blocks of different sizes

In order to determine the location, given by ($\boldsymbol{R_{i,x}}$, $\boldsymbol{R_{i,y}}$), we first start with a two-dimensional helper matrix, of size equal to the size of the original image, and whose elements are all initialized to 0. We denote this helper matrix as ***used***. In addition, we need to know the size of the smallest range block (stored in the header of the encoded file).

When we decode a range block, we obtain the corresponding size of that range block, and proceed to mark out all the pixels "occupied" by that range block in the ***used*** matrix. We then proceed to decode the next range block. The position of the next range block will be to the immediate right of the previously decoded range block. In other words, its horizontal position is the same as that of the previous range block, while its vertical position is given by the first unused pixel in the used matrix. If the rightmost end of the matrix is reached, we then move the vertical pointer down by the size of the smallest range block, and proceed to find the first unused pixel in the used matrix. As such, we are able to obtain the location of each range block ($\boldsymbol{R_{i,x}}$, $\boldsymbol{R_{i,y}}$).The positions of each range block are stored in matrices R_x and R_y.

4.2 Reconstructing the Image

With the seven matrices ***D***, ***D'***, ***T'***, ***I'***, ***LC'***, $\boldsymbol{R_x}$ and $\boldsymbol{R_y}$, the decoder now has sufficient information to reconstruct the image. For each range block, the decoder obtains the (x, y) location of that range block from $\boldsymbol{R_x}$ and $\boldsymbol{R_y}$, and regenerates the range block by applying the correct transformations to each used domain block in ***D***. Finally, with the reconstructed image, the decoder writes the data out in BMP format.

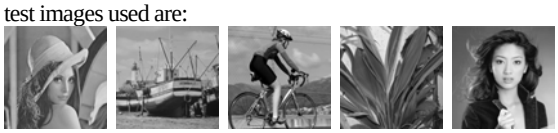
5. THE PSNR METRIC

In measuring the compression performance of the TIES algorithm, as well as that of the FIC algorithm, it is useful of have a precise and formal method of measuring image quality, rather than relying purely of visual inspection of the final decoded image. Hence, this paper makes use of the peak signal-to-noise ratio (PSNR) to represent image quality.

$$\text{PSNR} = -20 \log \frac{\sum_{p=0}^{total\ pixels} \left| image_{a,p} - image_{b,p} \right|}{total\ pixels \times 255} \text{ dB} \quad (2)$$

6. RESULTS AND DISCUSSION

In evaluating the performance of the TIES encoder, an empirical analysis was carried out on five grayscale test images. These test images were chosen as representatives of certain broad classifications of image types, which include human subjects, man-made objects and natural scenery. During the empirical analysis, two separate encodings, one by the TIES encoder, and the other by the FIC encoder [3], were performed on a total of four different image sizes – 128x128px, 256x256px, 512x512px and 1024x1024px – for each of the test images. The number of linear combination elements that the encoder was allowed to use was limited to four, as four elements tend to produce a suitable balance between image quality and image compression. The five test images used are:



Lena (1) Boat (2) Bike (3) Plant (4) Lady (5)

6.1 Compression Performance of TIES vs. FIC

In this section, the compression performance of the TIES encoder in comparison to the FIC encoder is examined. Based on the empirical results [5], with small images, the compression performance of the FIC encoder is better than the TIES encoder regardless of image quality (PSNR). However, when the size of the encoded image increases to 1024x1024px, the TIES encoder consistently produces better compression ratios compared to the FIC encoder at equal or better PSNR values. Table 6.1 below summarizes the specific improvement in compression performance of TIES over FIC for 1024x1024px images.

Table 6.1. Compression Performance Gains for TIES over FIC for large (1024 x 1024px) images

Image	PSNR Range (dB)	% Improvement in Compression over FIC		
		Min	Max	Mean
Lena	37 – 41	6.0	12.5	9.3
Boat	37 – 40	5.5	13.0	6.8
Bike	35 – 37.5	2.5	6.0	5.5
Plant	33 – 37.5	2.5	11.0	8.4
Lady	33 – 37.5	7.0	13.0	10.0

From the table above, we observe that within a given overlapping range of PSNR values produced by the TIES encoder and the FIC encoder, the TIES encoder achieves greater savings in the range of 2.5 to 13.0% over the FIC encoder. As such, this paper claims that while the TIES encoder produces smaller savings on average when compared to FIC for small-sized images, TIES will outperform FIC as image size is increased.

6.2 Savings due to Domain Compression (TIES)

In Fig. 6.1, the savings produced by the TIES encoder due to domain compression (only) is illustrated.

From the graphs above, two observations can be made. Firstly, for all six images, a reduction in domain pool size is achieved by compressing the domain pool. Furthermore, from the statistics captured, it appears that the size of the compressed domain pool is in the range of 1 to 11% of the size of the original domain pool.

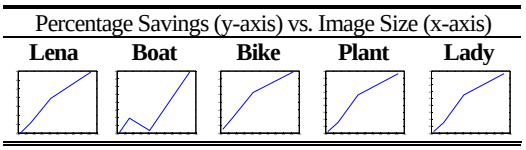


Fig. 6.1. Percentage Savings due to Domain Compression

Secondly, we observe that for the test images, generally the percentage savings due to domain pool compression increases as the size of the test image increases. As such, this paper suggests that there is a positive, linear relationship between the size of the original image and the percentage savings gained due to domain pool compression (excluding statistical fluctuations)

6.3 Savings due to Arithmetic Coding (TIES)

In Fig. 6.2, the savings produced by the TIES encoder due to arithmetic coding (only) is now shown for each image.

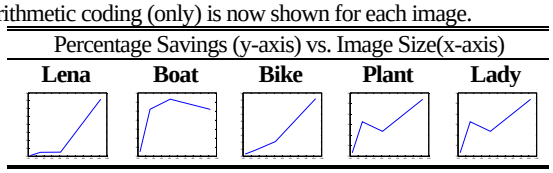


Fig. 6.2. Percentage Savings due to Arithmetic Coding

Again, we observe that the empirical results suggest that savings in the range of 15 to 26% can be achieved by encoding the data blocks using an arithmetic coder [6]. In addition, this paper again suggests that the relationship between the size of the image being encoded, and the percentage savings achievable due to arithmetic coding is linear and positively correlated. Indeed, this is in line with the results of the well-understood arithmetic-encoding algorithm, where data sets whose elements have asymmetric frequencies can be compressed significantly.

6.4 Encoding and Decoding Times of TIES vs. FIC

In Fig. 6.3, the average encoding and decoding time values for each image size, across all six-test images were taken for both the TIES encoder and the FIC encoder are shown. The horizontal axis of the graph represents the size of the image being encoded, while the vertical axis represents the time taken in seconds for the encoding process to complete.

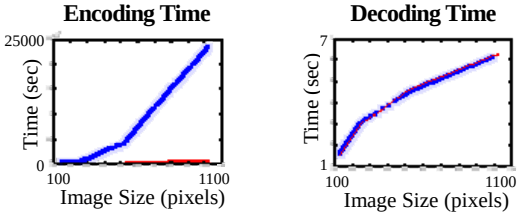


Fig. 6.3. Encoding and Decoding Times of TIES (thick) and FIC (thin)

It can be observed from Fig. 6.3 that for small images (256x256px and below), TIES and FIC complete the encoding process in approximately the same amount of time. As the image size increases, the graph above suggests that the encoding time for TIES increases much faster than that of FIC. However, the long encoding time for TIES can be reduced by using parallel processing techniques [7, 8, 9]. As for decoding, the decoding time for TIES is comparable to that of FIC.

Despite that the proposed TIES encoder produces better compression as compared to the FIC encoder as image size increases, a caveat is that the TIES encoder takes an extremely long time to encode a given image, and the encoding time increases extremely fast, as can be seen from Fig. 6.3, as the image size increases. A complexity analysis of the TIES encoding algorithm will point to the fact that the most computationally intensive segment of the TIES encoder lies in two components – the domain compression algorithm and the linear combination search algorithm. In particular, given d domain blocks and r range blocks, the linear combination search algorithm has to compute 2 flip, 4 rotation and 4 intensity transforms, and up to 4 linear combination elements for each d, which amounts to 128d domain block transformations, and $128dr$ range-domain block comparison. In addition, each range-domain block comparison operation runs in O (n^2) time. Furthermore, the number of domain blocks and range blocks are extremely large, where any given n x n pixel image has approximately $0.0625n^2$ range blocks and $0.25n^2$ domain blocks, if the smallest allowable block size is 4 by 4px. Hence, the estimated complexity of the linear combination search algorithm can be said to be of order n^6, which, while still being a polynomial-time algorithm, it clearly computation intensive and time-consuming relatively to the other parts of the TIES encoding algorithm. As such, in this paper, we propose to reduce the encoding time by means of parallelizing the TIES encoder. As such, this paper will describe a hardware architecture that will be used to target the parallel TIES algorithm, and the parallel TIES encoder itself.

7. THE PROPOSED PARALLEL ALGORITHM

As mentioned, the two most computationally intensive components of the sequential TIES algorithm are the domain compression algorithm and the linear combination search algorithm. In this section, we will proceed to parallelize these two components. Finally, the parallel algorithm utilizes the worker-coordinator paradigm of parallel programming, as will be discussed subsequently.

7.1 Target Hardware Architecture

Since the development of parallel algorithms is closely tied to the hardware architecture that the parallel algorithm is to run on, before developing the parallelized version of the TIES encoder, it is necessary to decide on a hardware platform on which to target. For the parallel TIES encoder, a distributed-memory MIMD architecture was chosen, since MIMD architectures are, by and large, one of the most general architectures and most non-specialized computer clusters in operation today are based on MIMD. This paper also chooses to use distributed-memory architecture since such architectures are easily and readily setup, and well-supported software libraries for developing such parallel applications, such as the Message Passing Interface (MPI), exist.

7.2 Parallelizing the Domain Compression Algorithm

We can see that when the algorithm reaches the point in which it has to compress the domain pool, the algorithm already has the complete domain pool and range pool. The algorithm then searches the entire domain pool for domain blocks, which are self-similar (upon applying some transformation), and are thus redundant. The algorithm then removes all such blocks from the domain pool. Clearly, since the entire domain pool is available to each process, the process of searching the domain pool can be easily sub-divided among the p processes available, with each process taking a range of domain blocks to check for redundancy. After each process completes its share of the workload, each worker process then sends the final set of non-redundant domain blocks to each other to form the complete compressed domain pool.

7.3 Parallelizing the Linear Combination Search Algorithm

In addition, in the sequential version of the linear combination search algorithm, for every range block in the range pool, the algorithm searches for a linear combination of domain blocks

such that:

$$R - \sum (D_1 + D_2 + D_3 + ... + D_n) \rightarrow 0 \qquad (3)$$

As such, it can be seen that the process of finding the linear combination of domain blocks that best represents a given range block is again independent of the search process for any other range block. Hence, a particular process will be able to obtain the linear combination of domain blocks for a given range block without the need for any communication, as long as it has available the domain pool of the given image.

7.4 The Parallel Algorithm

With this in mind, we proceed to develop the parallel algorithm. In the algorithm, there is 1 coordinator (root) process, and $w = p–1$ worker processes. However, the coordinator process is not a dedicated coordinator process, as it will also participate in the core computation of domain pool compression and searching for the best linear combination, as shown in Table 7.1.

Table 7.1. Sequences of Operations for Parallel Algorithm

No.	Root	Worker(s)
1	Read original image	Idle
2	Send original image to worker(s)	Receive original image from Root
3	Compute domain pool Compute range pool	Compute domain pool Compute range pool
4	Compress domain pool	Compress domain pool
5	Receive compressed domain pool from worker(s) and merge results	Send compressed domain pool to Root
6	Distribute combined compressed domain pool	Receive combined compressed domain pool
7	Linear Combination Search	Linear Combination Search
8	Receive search results from worker(s) and merge results	Send search results to Root
9	Output results	Idle / Terminate

In this algorithm, the workload of domain pool compression and linear combination searching is divided proportionally among the k processes (including the coordinator process), such that each process processes (d / k) domain blocks for the domain pool compression algorithm, and (r / k) range blocks, for the linear combination search algorithm. Once each process completes its allocated portion of work, it sends its results back to the root process, which combines the collective results to produce the final result, which would have been produced in the sequential algorithm. In the case of domain pool compression, the root process redistributes the combined compressed domain pool to all other workers, as this is needed in the linear combination search phase. However, such redistribution is not necessary after the workers send their linear combination search results back to the root process, as the root process simply has to write out the search results to the encoded file and terminate.

8. RESULTS AND DISCUSSION

8.1 Test Platform

The computer cluster used to test the parallel implementation of the TIES encoding algorithm is Tembusu2, a Unix-based 64-node cluster (quad-processor Pentium 4 Xeons) running the Linux operating system. Each machine is equipped with a total of 4 GB of memory. In addition, when running the parallel algorithm on the cluster, every process was run on a separate machine to as to ensure that communication overhead remains relatively constant (subject to network traffic and congestion), since running two processes on a single machine would result in a much lower communication cost as compared to running two processes on two individual machines, interconnected via a network.

8.2 Analysis and Results

The parallel TIES algorithm described in Section 7 was implemented and run against the same five test images described in Section 6. For each image, different sizes of the same image, ranging from 128x128px to 1024x1024px were used, and two measurements were taken – the total execution time of the parallel algorithm, and the total computational time of the parallel algorithm. Based on these two measurements, the speedup of the algorithm, relative to a given image, was computed. The results of the empirical analysis are:

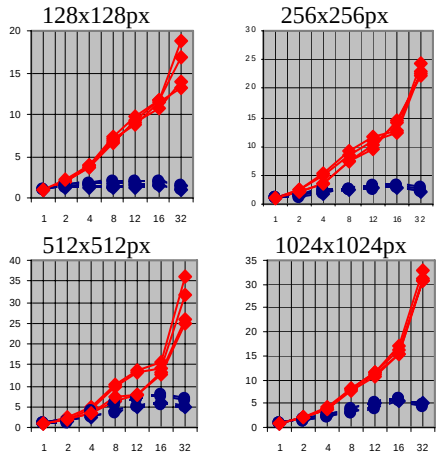


Fig. 8.1. Speedup (y-axis) vs. Processors (x-axis) Graph

The dotted line denotes the real speedup obtained, while the solid line denotes the computational speedup obtained. Here, we define real and computational speed as follows:

$$\text{Real Speedup} = \frac{\text{Total execution time on 1 process}}{\text{Total execution time on p processes}} \qquad (4)$$

$$\text{Computational Speedup} = \frac{\text{Total time spent computing on 1 process}}{\text{Total time spent computing on p process}} \qquad (5)$$

From Fig. 8.1, we observe that, in general, the real speedup for the total execution time across the five test images is relatively similar, as can be seen by the closely meshed lines. However, we note that for larger images (512x512px to 1024x1024px), the real speedup gained is more significant, in the range of 1.5 to 5.1, as compared to the smaller images (128x128px to 256x256px), of which the real speedup obtained is between 1.01 and 1.4. However, for each image size, there appears to be a limit to the real speedup obtained, as seen by the peak of the curve. For instance, for the 512x512px images, the maximal speedup obtained is in the range of 5.1 to 5.6, with 12 to 16 processes, while that for the 1024x1024px images is in the range of 5.61 to 5.7, with 16 processes. Further increasing the number of processes used yields a longer execution time (and thus a smaller speedup) as compared to using 12 to 16 processors. This is likely to be due to the overwhelming increase in communication overhead among the processes (to the root process), and vice versa, such that each process is doing too little work to justify the communication cost sustained. This will become more apparent after observing the computational time for each image size.

In order to eliminate the cost of communication overhead, which is highly dependent on the setup and implementation of the computing cluster used, the red lines (diamond markers) in Fig. 8.1 shows the computational speedup for each image size. Clearly,

in this case, each process will have a slightly different computational time associated with it, since some processes might have to search for a smaller average number of linear combination elements compared to some other process. As such, when measuring the computational time, the computational time of the longest running process is used.

From Fig. 8.1, we observe almost linear speedup with respect to the computational cost when running the sequential algorithm (shown using 1 processor). In addition, there is no observable limit to the speedup obtained, as increasing the number of processes to *k* leads to approximately *1/k* execution time. In fact, as can be seen the graph, in some cases a super-linear speedup is observed. However, this is likely to be due to statistical fluctuations in terms of the way the operating system schedules each process, such that at times a process runs faster (scheduled more frequently) than other times, and the fact that there is a probability that for a given number of processes, the load distribution may be more balanced than with another number of processes. As such, this paper suggests that the speedup obtained in terms of computational time is linear, since, as mentioned earlier, the nature of the domain compression algorithm and linear combination algorithm is such that the work performed by each process is independent, once the domain and range pool have been pre-computed.

Based on these observations, we can see that the main cost in the parallel algorithm is the cost of communication. There are two reasons for this. Firstly, on some parallel computing clusters, there exists two communication modes – (1) system level transfer mode, where node-to-node communication is handled entirely by the cluster itself with all communication channels and bandwidth being shared among all users, and (2) user level transfer mode, where a given user has exclusive access to the communication channel between two processors. On the Tembusu2 cluster, where the empirical analysis was carried out, only system level transfer mode is available. As such, due to the communication channels being heavily shared among users (more than 30 active users at the time of testing), communication cost is extremely high. Secondly, on the Tembusu2 cluster, the buffer size per channel (or pipe size) is relatively small, and this results in more communication being performed for large images to be transferred between any two processors. As such, this paper suggests that the less than ideal real speedup achieved is primarily due to the high communication cost of the test cluster. Hence, it is likely that the real speedup obtained would be much more significant if a low-latency network setup were used, or if the algorithm were run on a dedicated network of machines. Finally, if the algorithm is migrated to a shared-memory architecture, communication cost would decrease greatly and thus the algorithm is likely to yield much more significant speedups.

9. CONCLUSION

This paper has proposed an improvement scheme (Two-Pass Improved Encoding Scheme) based on the concept of fractal image compression. In describing the TIES encoder and decoder, three algorithms are used, which forms to basis of the TIES encoding algorithm – domain pool compression, best linear combination search, and block-based arithmetic coding. To exemplify the workability of TIES, and to demonstrate that under similar conditions, the TIES algorithm does outperform its FIC counterpart for large images of the size of 1024x1024 pixels and above, we have derived a complete implementation of the TIES encoder and decoder, and obtained empirical results based on six test images.

From the results of the empirical analysis carried out in Section 6, we observed that the TIES encoder produces good results in terms of gaining additional compression over the FIC encoder for large images of size 1024x1024px, in the range of 2.5 to 13.0% with respect to the FIC encoder for large images. Additionally, we observed that this gain in compression performance comes with a drawback – the encoding time is prohibitively long. As such, this paper proceeded to propose and implement a parallelized TIES encoder, to overcome the long encoding time. Based on the empirical results obtained from the parallelized TIES encoder, we observe that less than ideal speedup was obtained for the real encoding time, but almost linear speedup was obtained in terms of computational time, neglecting communication time. Hence, the parallelized implementation of TIES requires a low communication-cost environment, and since the Tembusu2 cluster consists of 16 quad-processor nodes, TIES is only suitable to run on the Tembusu2 cluster with a maximum of four nodes, under the system level transfer mode, so as to allocate a single quad-processor machine to running the TIES algorithm.

10. ACKNOWLEDGEMENTS

This authors' work was supported by the National University of Singapore Academic Research Fund under grant R-252-000-118-112.

11. REFERENCES

[1] M. F. Bansley, Fractals Everywhere, Academic Press, Inc., 1988.

[2] A. E. Jacquin, “Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations”, Institute of Electrical and Electronics Engineers Transactions on Image Processing, 1992, 18-30.

[3] Y. Fisher (editor), Fractal Image Compression – Theory and Applications. Springer-Verlag, New York, USA, 1995.

[4] S. G. Mallet, Z. Zhang, “Matching Pursuits with Time-Frequency Dictionaries”, Journal of Institute of Electrical and Electronics Engineers Signal Processing, December 1993, 41(12).

[5] K. W. Ching, G. H. Ong, “An Improvement Scheme to Fractal Image Compression – A Two-Pass Improved Encoding Scheme”, Proceedings of 11th National Undergraduate Research Opportunities Programme Congress 2006, National University of Singapore, February 2006.

[6] I. H. Witten, R. M. Neal, et al, “Arithmetic Coding for Data Compression”, Communications of the Association for Computing Machinery, 1987, 30(6): 520-540.

[7] J. Hammerle, A. Uhl, “Fractal image compression on MIMD architectures II, Classification based speed-up methods”, Journal of Computing and Information Technology (Special Issue on Parallel Numerics and Parallel Computing in Image Processing, Video Processing, and Multimedia), 2000, 8(1): 71-82.

[8] S. Lee, S. Omachi, et al, “A Parallel Architecture for Quadtree-based Fractal Image Coding”, and Proceedings of the 2000 International Conference on Parallel Processing, August 2000, 15-22.

[9] M. Wang, R. Liu, et al. “Adaptive Partition and Hybrid Method in Fractal Video Compression”, to appear in a special issue of Journal of Computers and Mathematics with Applications, 2006.

High Performance Computing Issues for Grid Based Dynamic Data-Driven Applications *

Craig C. Douglas
Computer Science Department, University of Kentucky
773 Anderson Hall, Lexington, KY 40506-0046, USA and
Computer Science Department, Yale University
P.O. Box 208285, New Haven, CT 06520-8285, USA
Email: douglas-craig@cs.yale.edu
And
Gabrielle Allen
Department of Computer Science and Center for Computation and Technology, Louisiana State University
302 Johnston Hall, Louisiana State University, Baton Rouge, LA 70803, USA
Email: gallem@cct.lsu.edu
And
Yalchin Efendiev , Guan Qin
Institute for Scientific Computation, Texas A&M University, Address,
612 Blocker, 3404 TAMU, College Station, TX 77843-3404, USA
Email: efendiev@math.tamu.edu and guan.qin@tamu.edu

ABSTRACT

DDDAS creates a rich set of new challenges for applications, algorithms, systems software, and measurement methods. DDDAS research typically requires strong, systematic collaborations between applications domain researchers and mathematics, statistics, and computer sciences researchers, as well as researchers involved in the design and implementation of measurement methods and instruments. Consequently, most DDDAS projects involve multidisciplinary teams of researchers.
DDDAS enabled applications run in a different manner than many traditional applications. They place different strains on high performance systems and centers due to dynamic and unpredictable changes in resources that are required during long term runs. An on demand environment is required. In this paper, we will also categorize many of these differences.

Keywords: DDDAS, HPC, Dynamic scheduling, Resource allocation, Data centers, Data filters, Data assimilation.

1. INTRODUCTION

Dynamic data-driven application simulation (DDDAS) is an emerging field of science and technology [1-6]. Developing a symbiotic two way interaction between (intelligent) sensors and multi-model, multi-scale simulations provides a more accurate methodology. Implementing DDDAS, instead of the traditional take an initial guess and run a simulation for some short period of time, has impacts on high performance computing that affects the following:

1. Scheduling and allocations
2. Infrastructure (including visualization, computer environment and performance, and communication)
3. Middleware
4. Data integrity

* This paper was supported in part by National Science Foundation grants CNS-0540178, CNS-0540136, ACI-0324876, and CNS-0540374.

DDDAS is an example of on demand everything. In order to actually implement DDDAS correctly and precisely, we need to be able to change all high performance computing (HPC) parts of the computation right now, again later, and we cannot define everything in advance.

When DDDAS works well, it assumes that almost everything can be modified during the course of a long term simulation. The diagram in Fig. 1 shows how a number of elements might interact with each other: All six of the components may change without resorting to a new simulation as the computation progresses. Virtually all DDDAS applications are multiscale in nature. As the scale changes, models change, which in turn, changes which numerical algorithms must be used and possibly the discretization methods. To date, almost all DDDAS applications involve a complicated time dependent, nonlinear set of coupled partial differential equations, which adds to the complexity of dynamically changing models and numeric algorithms. It also causes computational requirements to change, particularly if dynamic adaptive grid refinement or coarsening methods are used. This is not something normally supported at supercomputer centers, but is slowly being added.

2. SCHEDULING

DDDAS impacts scheduling in three ways directly:

1. Model-model coupling across wide area networks
2. The controlling computational device
3. Immediate, dynamic on demand scheduling

Model-model coupling can be quite complicated to implement. Besides mathematical modeling issues there is a HPC aspect that is subtle. Each model may require substantial computational resources to be effective. Each model may be owned by a specific entity that requires that its code must be run at a specific location. In both cases, the models may be run in different locations, separated by

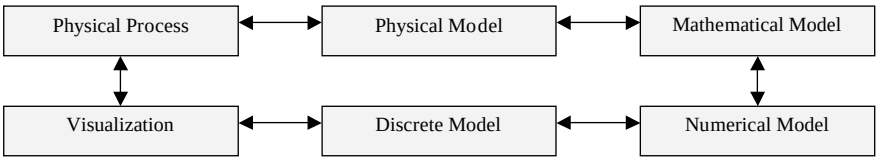


Fig. 1. DDDAS processing.

significant distances.

The network then must be scheduled if substantial amounts of data will be moved between the different locations. There is an issue of high speed, high latency versus low speed, low latency networks. Worse, there is usually no way to control how the data moves between data centers. One ancient, low speed Ethernet cable somewhere in a building can intermittently ruin high speed data transfers if a router occasionally and consistently uses it.

Co-scheduling of both computational resources becomes necessary when different computer centers are involved. Additionally, the network between the sites may have to be scheduled since there are only a maximum number of bits per second that can be transferred across the network connecting the sites. When more than two sites are involved or a regular, public network is involved (versus a research, monitored, and scheduled network), things get really interesting.

The controlling computer in a simulation is not necessarily a fixed workstation with a fixed network address. It is becoming imperative to support mobile devices (e.g., a laptop, PDA, or cell phone) as the controlling device. These devices move and must be able to reconnect to a simulation, be findable when connected to a network, and to robustly run complicated simulations with similar devices coming online and going offline randomly during a computation. Some device on the network typically acts as a broker in order to keep track of all of the sensors and computers involved as well as to provide security to the DDDAS.

Examples include a wildland fire, a burning building, or moving water contaminants. The sources of information are not fixed. The person responsible for coordinating a disaster management needs to be in the area to see what is happening (and not necessarily in a fixed location). Techniques developed in point to point (P2P) are useful here. The controller may be a separate device controlling a Grid of devices (sensors and computers). As the device moves, it may change networks, drop out of connectivity, and must be able to rejoin the DDDAS seamlessly.

The DDDAS may require dynamic and immediate on demand scheduling. Since there is a symbiotic relationship between the computational kernels and the sensors, the sensors may indicate to the DDDAS that different mathematical or physical models need to be used. The need may be immediate to continue processing. The computational components corresponding to the request may be on different machines in the Grid and have to be run on demand.

An example is a movable drone that can detect different chemicals in water. If it detects certain petroleum products in water, there are two interesting possibilities: a recent boat sinking, just a leaky fuel tank on a functional boat, or a major (possibly intentional) fuel spill. Checking for further chemicals and concentrations will determine which case exists. One of three different computational models will run after the decision and the sensor may have to be reprogrammed to continue functioning usefully as part of the DDDAS.

The computational components may not be in the same location. However, we need resources immediately and we may be dealing with batch queues. We need either fast injection or pre-injection of the application into the queue.

Fast injection into the queue means that the batch system has the ability to flush the queue quickly. It must be able to write running applications to disk very quickly, which is a serious problem disk since writing many gigabytes of data to disk is slow. In addition, most operating system kernels do not have the ability to roll in a new application while rolling out the old ones in a fast manner. This requires a kernel change, which several computer vendors are trying to implement.

Pre-injection into the queue requires starting an application and then putting it to sleep without allocating any memory for data. The application's footprint in the computer is small and as long as the batch system does not terminate the application for lack of using CPU cycles, restarting it is fast. Ideally the application uses a small amount of memory per processor (e.g., as would be done on machines like an IBM BlueGene/L or an old Intel Paragon). By using more processors, but less memory per processor, we save paging time when restarting and can, thus, be useful to the DDDAS quicker.

3. INFRASTRUCTURE IMPACT

Information about the hardware and software environments is needed to make a DDDAS run efficiently on the available resources. Standard application programming interfaces (APIs) are essential. Creating a standard DDDAS API is an ongoing research area and is beyond the scope of this paper.

The APIs are necessary so that data structures, load balancing and computational algorithms can be dynamically employed. Thus, a DDDAS can run efficiently and not waste resources or time.

Having consistent dynamically linked libraries is essential, yet one of the largest causes of failure to run applications on a Grid. Using the same variant of Linux, for example, is insufficient to guarantee that the libraries are consistent and interoperate. The lack of consistency leads to a very subtle forms of failure that are difficult to track and fix during a DDDAS that must be run during a specific period of time.

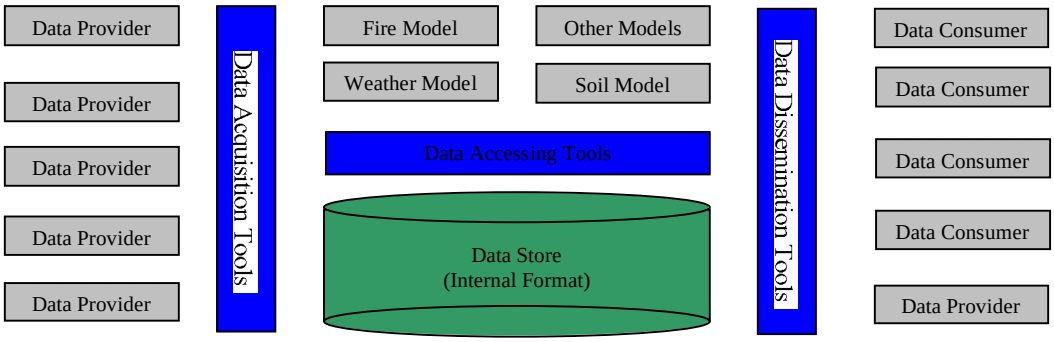


Fig. 2. Data acquisition, accessing, and dissemination software layout in a typical DDDAS project (e.g., a wildland fire DDDAS project).

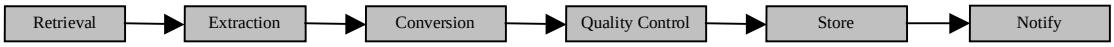


Fig. 3. A General View of Data Acquisition Tools.

The APIs need to be able to help determine when and when to stage executables. We need to know how much allocation time is remaining for specific batch queues so as to prioritize work in these queues. We need to know the queue wait times (before execution begins) to prioritize the locations for scheduling work. Finally, we need the ability to find available hardware resources that we might not normally utilize that are idle.

DDDAS also impacts performance. Using standard APIs like PAPI, the applications can dynamically optimize themselves using standard methods (though painful to write initially). Parameters can also be chosen dynamically, including the frequency of analysis of the overall application's effectiveness.

APIs are needed to negotiate increases or decreases of allocations on the computers in the Grid. The most common causes of workload changes are the following:

1. Dynamic adaptive mesh refinemen and coarsening.
2. Changing the model (e.g., character of the governing equations)

Cyber security has a major impact on DDDAS. Firewalls cause problems starting with data collection and transfers to application transfers to other computer sites on the Grid based on decisions made with respect to available queues. Firewall security levels may change dynamically at some sites based on network flow analysis, which may be effected by the DDDAS itself.

Port blocking is probably the single hardest factor to accommodate. Data may traverse long distances and multiple networks. Anywhere along the routing, a network administrator can arbitrarily block a port for no apparent reason and without notifying anyone. Suddenly the entire DDDAS stops working. Finding the source of the problem is difficult and nontrivial to get fixed.

Both cyber security issues are current research areas for the DDDAS community.

Most DDDAS need real time visualization at some point. We need to be able to provision high speed networks in order to see what the actual bandwidth is. We need flexible scheduling so that high priority visualization can occur when absolutely necessary. We also need to be able work with radically different environments at the same time, e.g., immersive three dimensional devices, a simple LCD screen, and a small screen on a PDA or mobile phone.

4. DATA INTEGRITY

To support DDDAS' requirements, data acquisition, data accessing, and data dissemination tools are typically used. Data acquisition tools are responsible for retrieving of the real-time or near real-time data, processing, and storing them into a common internal data store. Data accessing tools provide common data manipulation support, e.g., querying, storing, and searching, to upper level models. Data dissemination tools read data from the data store, format them based on requests from data consumers and deliver the formatted data to the data consumers. Fig. 2 illustrates a simplified view of a typical DDAS system.

The data used to drive a DDDAS system are retrieved periodically by a data retrieval service, extracted, converted, quality controlled, and then save to the internal data store. After the data is stored, the data accessing tools are notified which can then update the input to the simulation models. The whole process is illustrated in Fig. 3. The extraction process reads the retrieved data based on the meta data associated with them and feeds the extracted values to the conversion model whose major purpose is unit conversion, e.g., from inches to millimeters. The converted data are then analyzed for potential errors and missing values by the quality control model. This control process will ensure the correctness of the data, which is of great importance for the model simulation accuracy. The quality controlled data are then fed to the data storage model, which either saves the data to a central file system or loads them to a central database (this depends on project requirements). The data store model may also need to register the data in a metadata

database so that other models can query it later.

DDDAS projects are highly multidisciplinary in nature and are developing comprehensive IT tools, mathematical models, and prototype infrastructure for disaster modeling and management. The projects will bring comprehensive information and numerical prediction where it is needed, at the disaster command center, in real time.

5. CONCLUSIONS

DDDAS is an oncoming force in computational science. The HPC community is ill prepared for it. Basic information technology research is necessary in order to accommodate DDDAS on a Grid. DDDAS will impact the HPC community through middleware to resolve new paradigms for dynamic scheduling, allocations, resource provisioning, data centers, and cyber security..

6. REFERENCES

[1] http://www.dddas.org. Includes project descriptions, many DDDAS workshop virtual proceedings, and links to DDDAS software.

[2] K. Baldridge, G. Biros et al, "January 2006 DDDAS Workshop Report", National Science Foundation, 2006, http://www.dddas.org/nsf-workshop-2006/wkshp_report.pdf.

[3] P.M.A. Sloot, D. Abramson et al "Lecture Notes in Computer Science, Vol. 2660", 2003 Dynamic Data-Driven Application Workshop, F. Darema, ed., in Computational Science - ICCS 2003: 3rd International Conference, Melbourne, Australia and St. Petersburg, Russia, June 2-4, 2003, Proceedings, Part IV, Springer-Verlag Heidelberg, 2003, 279-384.

[4] M. Bubak, Geert Dick van Albada et al, "Lecture Notes in Computer Science series, vol. 3038", 2004 Dynamic Data-Driven Application Workshop, F. Darema, ed., in Computational Science - ICCS 2004: 4th International Conference, Kraków, Poland, June 6-9, 2004, Proceedings, Part III, Springer-Verlag Heidelberg, 2004, 662-834.

[5] V.S. Sunderam, Geert Dick van Albada et al, "Lecture Notes in Computer Science series, vol. 3515", 2005 Dynamic Data-Driven Application Workshop, F. Darema, ed., in Computational Science - ICCS 2005: 5th International Conference, Atlanta, Georgia, USA, May 22-25, 2005, Proceedings, Part II, Springer-Verlag Heidelberg, 2005, 610-745.

[6] V.N. Alexandrov, G.D. van Albada et al, "Lecture Notes in Computer Science 3993", 2006 Dynamic Data-Driven Application Workshop, F. Darema, ed., in Computational Science – ICCS 2006: 6th International Conference, Reading, UK, May 28-31, 2006, Proceedings, Part III, 2006.

[7] C. C. Douglas, A. Deshmukh et al, "Dynamical data driven application systems: Creating a dynamic and symbiotic coupling of application/simulations with measurements/ experiments", National Science Foundation, Arlington, VA, 2000. http://www.dddas.org/nsf-workshop-2000/worksop_report.pdf.

The Comparisons Of Data Transfer Methods In An Ogsa Based Grid System

Zhili Cheng, Zhihui Du, Suihui Zhu, Man Wang
Department of Computer Science and Technology, Tsinghua University
Beijing, 100084, China
Email: chengzl02@mails.tsinghua.edu.cn

ABSTRACT

The performance and the ways of data transmission among different kinds of services are critical in an OGSA based grid system. Four widely used data transmission methods with different kind of transmission protocols and different operating mechanisms are provided. Based on the proposed model of one service invoking procedure with different kind of data size, some theoretical analysis results are given. At the same time, in a grid environment built up with Globus Toolkit 3, a set of experiments have been designed and tested. The theoretical results are compared with the experimental results. The suggestion put forward by the end of the paper would be helpful for most grid applications which demand of data transmission in different scenarios.

Keywords: Data Transmission, OGSA (Open Grid Services Architecture), Grid, GridFTP, Performance Comparison.

1. INTRODUCTION

Open Grid Services Architecture (OGSA) [2] is a framework, which provides a coupled, scalable and resilient distributed computing environment. Most of the grid systems conform to the OGSA specification, which abstracts different elements in the Grid system as services.

Since the Grid system is made up of services and increasing scientific and e-business applications have been developed in forms of grid service. The grid developers pay more attention to find a proper way for the services to communicate or exchange data with each other. So how to deal with the data exchanging among those services has become a very important problem.

This paper focuses on the analysis of different kinds of data exchanging technologies and how to choose suitable data exchanging technology for different kinds of applications. Section 2 summarizes four methods of data transfer in a Grid system by describing these methods' operational mechanisms. In Section 3, a universal model for complex data transfer among many services is proposed. In Section 4 and Section 5, the theoretical analysis of the performance of different methods has been made. We analyze these methods' performance by modeling the transfer time of each method with a few factors. A formula is given to each method to compute the theoretical transfer time. A set of data with different size is placed into the formulae. Based on the calculation results, some suggestions are given for different data exchanging scenarios. Afterwards, a set of experiments has been done to test the performance of the methods and the experiment results have been compared with the theoretical results. In Section 6, a conclusion is given to grid users and developers; it would be a guide for choosing proper transfer method in different application areas in different conditions.

A service execution pattern with a central scheduler is employed. A service can be both active and passive, which means the service can call other services and be called by others. But nowadays, in the applied field, most of the services are passive. A scheduler was introduced to release every single service from knowing other services' locations and invoking handles. This is sound for application developers who want to combine services and fulfill collaboration. As we analyze later in Section 4, the bottleneck of transfer time doesn't lie on the communication between services and scheduler, so if the scheduler was removed, the analysis and the experiments in this paper would also be helpful.

* This paper is partly supported by National Natural Science Foundation of China (No.60503039), Beijing Natural Sciences Foundation (No.4042018) and China's National Fundamental Research 973 Program (No. 2004CB217903).

2. DATA TRANSFER METHODS BETWEEN TWO SERVICES

In this section, data transfer models with different transfer methods are discussed. They are classified based on their transfer protocols or structures. For each method, we give a model sketch and a description of the operational mechanism.

2.1. Transfers by SOAP

SOAP [9] is a protocol base on XML. It defines the standard data transmission between web services. Since SOAP is based on XML, all of the data to be transmitted are transformed in an XML style. The whole process of dealing with XML is: the client serializes the data to be sent; the client encodes an XML (a SOAP message); the client sends it; the server receives it; the server parses (decoding) the received XML (a SOAP message) and un-serializing the XML factor to a real data. As a result of the extra process of XML, the efficiency of this method goes lower than RMI or transfer data files directly when the data size is large (We could see this in Section 4).

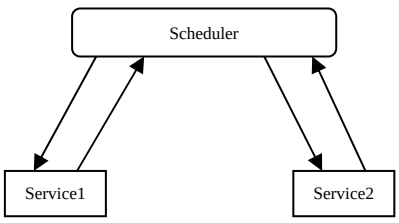


Fig. 1. Transfer by SOAP

Fig. 1 shows the method, which transfers data by SOAP between two services. The executing steps in this method are:

1) The scheduler calls service1 and sends the data that service1 needs in a SOAP message.
2) Service1 gets the data it needs, calculate with it, and return the result to the scheduler, of course, also in a SOAP message.
3) The scheduler invokes service2 with result received from service1; the procedure is similar to step 1.
4) Service2 does what Service1 did in step 2.

2.2. Transfers by File Servers and Service Called with Address Parameters

Since SOAP performs low efficiency in data transfer, a traditional way is considered to solve the problem. SOCKS communication would meet the needs, hereby we use Ftp to transfer large amount of data in practical application.

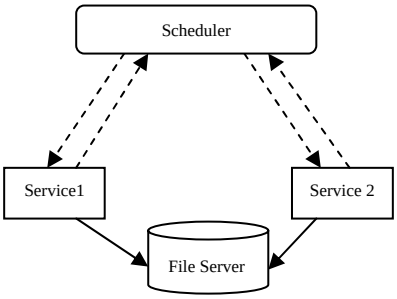


Fig. 2. Transfers by File Servers and Service Called with Address Parameters

Fig. 2 shows the method which uses one Ftp server as temp data transition media. The executing steps in this method are:

1) The scheduler invokes service1 and offers an address, where service1 could get data from.
2) Service1 downloads data from File Server (the scheduler has told its address in step 1), performs operations with the data, and uploads the result files to File Server, returns the address of result to the scheduler.
3) The scheduler gets the data address and directly calls service2 by providing the address of data source.
4) Service2 downloads the data it wants, does operations and uploads, returns the address to the scheduler.

2.3. Transfer by GridFTP File Servers and Service Called with Address Parameters

Ftp is a traditional way to transfer files through network, in Grid world, people developed a new method to do the transportation job. That is GridFTP [1], with parallel transfer ability, better security and stability.

Fig. 3 shows the method, which uses GridFTP servers as temp data transition media. The executing steps in this method are:

1) The scheduler invokes service1 and gives it a GridFTP address, where service1 could get data.
2) Service1 accesses the GridFTP servers, this time with parallel access and multiple TCP streams. After getting the data and dealing with it, service1 re-accesses the GridFTP servers; upload the result files with multiple streams; and return the address to the scheduler.
3) The scheduler gets the data address and directly calls service2 by telling where to get the data source.
4) Service2 does things similarly service1 in step1 did.

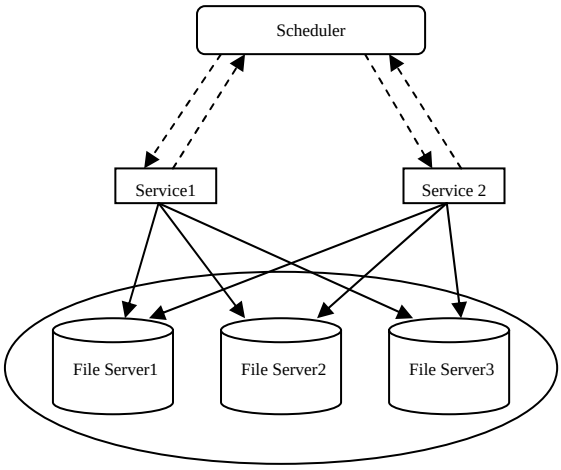


Fig. 3. Transfer by GridFTP Servers and Service Called with Address Parameters

2.4. Transfer by Self File Servers and Service Called with Address Parameters

At the end of this section, we present a method with local Ftp server. Here, local server means each service server is also an Ftp server. Each service's output data could be stored directly to the local server, which saves time of transfer through the network. This will bring high performance in transmission.

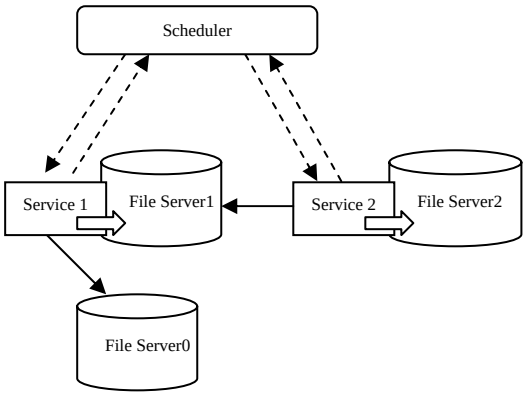


Fig. 4. Transfer by Self File Servers and Service Called with Address Parameters

Fig. 4 shows the method, which transfers files by local Ftp servers and invoke with address parameters method. The executing steps in this method are:

1) The scheduler invokes service1 with data address.
2) Service1 gets the address and gets the data from file server, process on it, and store the output data to the local server (this time it could store it directly by disk I/O, not through net), after that, service return the address of the output data to the scheduler.
3) The scheduler invokes service2 with the address from service1.
4) Service2 does things as service1 did in step2.

3. A DATA TRANSFER MECHANISM AMONG

MULTI-SERVICES

The data transfer among multi-services includes the following cases: a service's output is accessed by many services; many services' outputs are accessed by one service; and the combination of the two situations above.

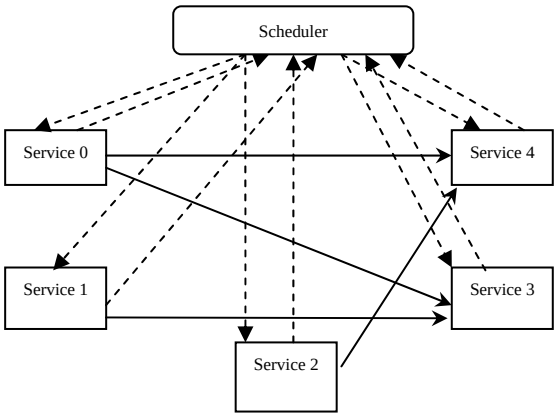


Fig. 5. Data Transfers among Multi-Services

To deal with this problem, we define a model called MM, which stands for "Multi to Multi". It means the service in the process may have many succesors and may also have many predessors. In this situation, things become more complicated. Here, we also employ the execution pattern with a central scheduler. All the interactions of services would pass through the scheduler. The pattern with a central scheduler is simple to implement but it is not an essential request.

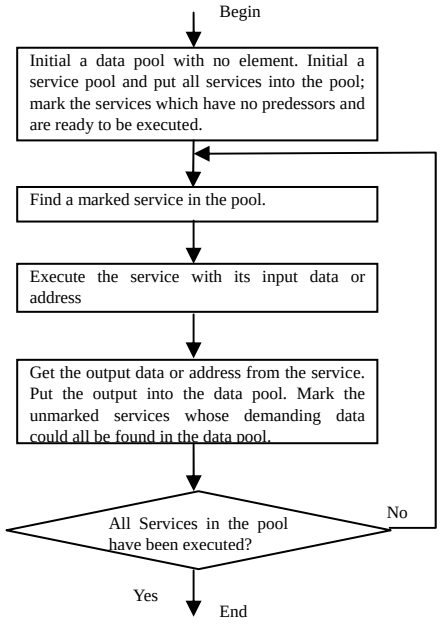


Fig. 6. Flow Chart of the Algorithm for MM's Data Transfer

Fig.5 gives an example of data transfer among multi-services. To explain the operating mechanism details of this model, a service pool and a data pool are introduced. Initially, make the data pool empty and put all the services in the service pool. Each service has its previous data demand (may be none). The scheduler searches in the service pool and finds a service which is ready to be executed (which means this service doesn't need data or the data needed are already in the data pool), executes the service with the initial data or address and puts its output to the data pool. Then repeat the following operations: finding a ready service whose input data are all in the data pool, executing it with the data or address, and putting the output data or address into the data pool. The procedure is showed in Fig. 6. The expression ability of this algorithm is equal to a DAG (Direct Acyclic Graph).

In Section 2, we have already considered the possible methods of transmission, and any of them could be used as the transfer method for the complex model described in this section.

4. THEORETICAL ANALYSIS AND COMPARISONS OF TRANSFER METHODS

First, the access types and transmission times of the methods we mentioned in Section 2 is given.

We use A, B, C and D to present for the 4 methods discussed in Section 2:

A. "Transfers by SOAP"

B. "Transfers by File Servers and Service Called with Address Parameters"

C. "Transfer by GridFTP File Servers and Service Called with Address Parameters"

D. "Transfer by Self File Servers and Service Called with Address Parameters"

Table 1 shows the of main transfer operations of each method in one transmission process. This table comes from the sketches in Section 2.

Table 1. the Comparison of Main Transfer Operation Types and Times of Transferring

Model	Transfer Type				
	Transfer data by SOAP	Transfer address by SOAP	Transfer data by Ftp	Put data to Grid FTP servers	Get data from Grid FTP servers
A	2				
B		2	2		
C		2		1	1
D		2	1		

In order to Model the transfer time for each method, we must find out which factors affect the transfer time. In our analysis, there are four factors: the time to transfer the data *T1*, the time to write data to local external storage *T2* (only for file transfer methods, not necessary for "transfer by SOAP"); data encode and decode time *T3* (only for "transfer by SOAP") and additional time *T4* (such as connection built up time and security check time). So the total transfer time *T* is shown down:

$$T = T1 + T2 + T3 + T4 \qquad (1)$$

Assume the transfer speed for Ftp server is 10M Bytes/s

(on 100M Ethernet). The GridFTP's transfer speed is also 10M Bytes/s. research in [3] shows that the transfer speed for data in forms of SOAP is about half of the Ethernet Bandwidth, which is about 5M Bytes/s. The hard disk I/O speed is about 50M Bytes/s. Assume the GridFTP system has an extra consuming time for security check which is about 100ms per access. Assume the file server connect time is about 50ms. Each transfer with an accessional time is about 50ms. Considering the XML parse and data serialize speed, which is about 1M Bytes/s. With these factors, Eq. (1) can be replaced with Eq. (2).

$$T = T_1 + T_2 + T_3 + T_4 = \frac{TransferTimes \times DataSize}{NetBandwidth} + \frac{StoringTimes \times DataSize}{StorageBandwidth} + \frac{ParseTimes \times XMLSize}{ParseEfficiency} + SystemComsuingTime \quad (2)$$

The formulae below show detail factors of each method. They are deducted from Eq. (2).

A:

$$\frac{2 \times DataSize \times 1000}{5MB/s} + \frac{2 \times DataSize \times 1000}{1MB/s} + 2 \times 50$$

B:

$$\frac{2 \times DataSize \times 1000}{10MB/s} + \frac{DataSize \times 1000}{50MB/s} + (2+2) \times 50 + ServerConnectTimes \times 50$$

C:

$$\frac{2 \times DataSize \times 1000}{10MB/s} + \frac{DataSize \times 1000}{50MB/s} + (2+1+1) \times 50 + ServerConnectTimes \times 50 + 2 \times 100$$

D:

$$\frac{DataSize \times 1000}{10MB/s} + \frac{DataSize \times 1000}{50MB/s} + (2+1) \times 50 + ServerConnectTimes \times 50$$

We analyze the methods with data in different sizes: 10KB, 100KB, 1MB, 10MB and 100MB. Put them into the formulae shown up, and the result is shown in Table 2: (The time unit is millisecond)

Table 2. Theoretical Analysis Results of Transfer Time in One Transmission Process of Each Method

	10KB	100KB	1MB	10MB	100MB
A	124	340	2500	24100	240100
B	302.2	322	520	2500	22300
C	502.2	522	720	2700	22500
D	201.2	212	320	1400	12200

From the formulae and the data in Table 2, it is clear that these time models present a cluster of lines with different slopes and initial factors. It shows that: If we want to find out which method is better in performance by giving a certain kind of data size in the theoretical framework, we could just draw out these lines and judge them directly.

5. EXPERIMENTS

In order to validate the theoretical analysis in Section 4, and evaluate the performance of the methods we describe above, some experiments are given in this section.

5.1. System Setup

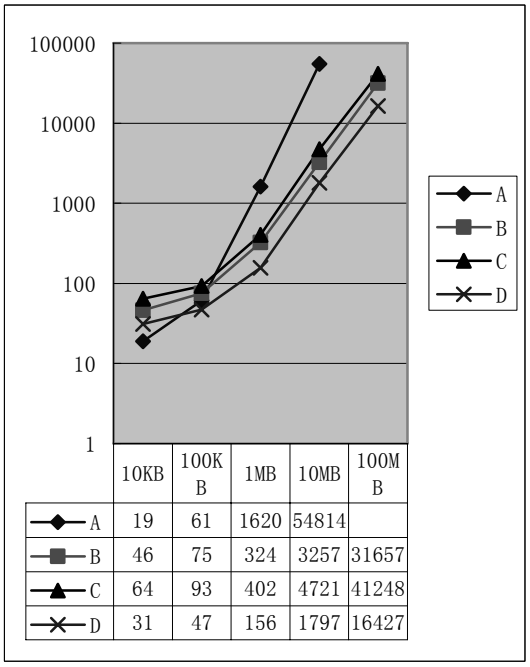
The hardware configuration comes to: four personal computers are used in the experiments, one as a central scheduler, one as an Ftp and GridFTP server and the other two as the service providers. The main hardware configuration of these PC is: AMD 1800+, 1GB DDR memory and 80GB 7200rpm hard disk. They are connected by 100M bandwidth Ethernet.

The software environment includes: GT3 (Globus Toolkit 3) [8] is used as the grid developing toolkit and platform [4]. The system is running on Windows XP + SP2, Java 1.4.1. Serv-U 6.0 is used as the Ftp server application, while Jakarta Commons/Net [6] is used to develop the Ftp client. When GridFTP server is in use, we turn to a Linux OS: Fedora Core 3, which also includes GT3, Java 1.4.1 and the operating environment, such as ANT etc. For the GridFTP client development, CoG Kit 4.1.3 [7] is used [5].

5.2. Experimental Method

The experimental method follows the operating mechanisms described in Section 2 and the data size partition described in Section 4. The data to be transferred are integers in forms of string; they are generated by java.util.Random. We use java time API to get the whole transfer process. For the transfer methods transferring each size of data, one experiment has been done for 10 times and an average of result is given as a final result.

5.3. Results and Performance Comparison



	10KB	100KB	1MB	10MB	100MB
A	19	61	1620	54814	
B	46	75	324	3257	31657
C	64	93	402	4721	41248
D	31	47	156	1797	16427

Fig. 7. Experiment Results and Comparison

Fig.7 shows the experiments results (The time unit is millisecond). The chart shows that the analysis gives the right trend of the performance of different kinds of data transfer methods. There is a vacancy in the table; this is

because, when data size reached 100M bytes, the transfer by SOAP method didn't work. It threw an "Out Of Memory" Exception.

The theoretical results and experimental results are compared with each other below.

First, when the data size is small, such as 10KB and 100KB, the predicted time is much bigger than the real ones, there may be two reasons: 1. In the theoretical analysis, when the data size is small, the main consume time is the additional system executing time, which is not easy to correctly estimate; 2. We test this system in Windows and use java timing API to judge the transfer time, which may not be precise, especially when it comes to millisecond level. Luckily, the absolute values are very small.

Second, when the data size becomes larger, the predicted time is smaller than the real ones. The reason for this comes to: in this condition, the transfer speed becomes the main factor which affects the whole process, and the Ftp transfer speed in the experiment didn't reach 10M Bytes/s, it's about 7M Bytes/s to 8M Bytes/s.

Third, we found out that, when data size becomes large (More than 1MB), the performance of the transfer method that transfers data by SOAP is not near linear. The bigger data it transfers the lower efficiency it performs. The results depend on the implementation of given XML parser. With another parser, the result might be different. But it convinces one principle: when the data size is small, we can use XML directly to transfer the data, but when the data size becomes big, we have to try other ways.

6. CONCLUSION

In this paper, different data transfer implementation methods are discussed. According to the theoretical analysis and the experiment results, a conclusion of the each method's suitable applied scenario is given below.

1) For simple e-business applications, especially towards the small enterprises, it is better to transfer data by SOAP directly. Data in these procedures is usually small in size (less than 100KB) and simple in type (usually some tables or documents made up of strings and numbers).

2) For scientific computations, especially for the condition that all of the hosts are in a restrained area (i.e. in a LAN), it is better to use the "Transfer by Self File Servers and Invoke with Address Parameters" method. It is fast, meanwhile its protocols are mature and its implementation is simple. If all the machines could not have an Ftp server, "Transfers by File Servers and Service Called with Address Parameters" is also a good choice.

3) For large corporations' e-business applications and distributed scientific computations with high security requirement, GridFTP will be a better choice. It provides secure, parallel and striped transfer, which is not included in Ftp. From the experiments, we could conclude that GridFTP still has a near-linear efficiency and it transfers a mass of data with a reliable mechanism.

With this conclusion of simple and usually used data transfer methods in OGSA based Grid systems, developers would make a proper decision when they want to transfer data in Grid systems, and they will gain more benefit from the Grid.

7. REFERENCES

[1] GridFTP: Protocol Extensions to Ftp for the Grid. http://www-fp.mcs.anl.gov/dsl/GridFTP-Protocol-RFC-Draft.pdf.

[2] I. Foster, C. Kesselman, et al, "The Physiology of The Grid, Grid Computing – Making the Global Infrastructure a Reality", ed. F. Berman, A. Hey and G. Fox, John Wiley & Sons, Chichester, England, 2003.

[3] S. G. Shasharina, N. Wang, et al, "Grid Service for Visualization and Analysis of Remote Fusion Data", Proceedings of the Second International Workshop on Challenges of Large Applications in Distributed Environments, 7 June 2004, 34 - 43.

[4] The Globus Toolkit 3 Programmer's Tutorial. http://gdp.globus.org/gt3-tutorial/multiplehtml/index.html

[5] V. Silva, "Transferring files with GridFTP". http://www-128.ibm.com/developerworks/grid/.

[6] http://jakarta.apache.org/commons/net/.

[7] http://wiki.cogkit.org/index.php/Main_Page.

[8] http://www.globus.org/toolkit/.

[9] http://www.w3.org/TR/soap/.

Automatic communication code generation in parallel compilation system

Xue-rong GONG1, Yong-hong SHENG2, Ping ZHANG3
National Digital Switching System Engineering &Technological R&D Center
Zhengzhou, Henan CHINA
E-mail: gongxuerong@163.com
2E-mail:xdsyh@126.com
And
Lin-sheng LU
Jiangnan Institute of Computing Technology
Wuxi, Jiangsu CHINA

ABSTRACT

The paper introduces computation decomposition and dependence relation in parallel compilation, and presents how to generate communication code automatically according to data and computation decomposition as well as dependence relation. The communication code consists of four parts: data distribution, communication before computation, synchronization communication, and data gathering. By founding different linear inequalities system and using FME method, we can create communication loop nest and insert MPI send and receive code into the loop nest. One linear inequalities system, used to create the code of data distribution and data gathering, is made up of data decomposition and the bound of the array. Another linear inequalities system, used to create the communication code before computation and the synchronization communication code, is made up of computation and data decomposition, data dependence relation, and the bound of iteration. This paper also presents several methods of communication optimization.

Keywords: parallel compilation, linear inequalities system, code generation, LWT, communication optimization.

1. INTRODUCTION

High-performance computing is rapidly becoming an important component of scientific research and development. Today's scientists and engineers depend on supercomputers and fast workstations to solve many important problems in many fields such as aeronautics, physics, biology, and medicine. Though speed of microprocessor is continuing to increase, most observers agree that parallel computing is the only way to significantly increase the computational power available. But each parallel machine has its own machine-specific features that must be considered in order to achieve efficient use of the underlying hardware. So programming for parallel architectures is a very burdensome task, since one has to take into consideration all special characteristics of the underlying hardware. Moreover, there exist lots of ready-made serial applications in every domain. If we run those serial programs on the parallel machine directly, the real performance may be far from the peak performance of the parallel platform. To overcome this difficulty, significant research effort has been aimed at developing source-to-source parallelizing compilers for multi-computers that relieve the application programmer from the task of program partitioning and communication generation.

In recent years, a lot of research has been done concerning parallel compiler, both in theory and practice. Some compiler infrastructures, such as UIUC's Po1aris (University of Illinois at Urbana—Champaign) [1]，Stanford University's SUIF (Stanford University Intermediate Format) [2] ， FUDAN University's AFT(Automatic Fortran Transformer) [3] and so on are also available.

The shared memory and distributed memory programming models are two of the most popular models used to transform existing serial application codes to the parallel form. Extensive discussions and practical experience have led to the conclusion that shared memory machines, although easy to program, lack in scalability, while on the other hand distributed memory machines are much more scalable but are definitely more difficult to the programmer. We aim to implement a tool which automatically generates the message-passing code from the serial code for distributed memory machines.

We use the SUIF compiler infrastructure to implement our compiler --- Kit of Automatic Parallel (KAP). The SUIF compiler consists of a small, clearly documented kernel and a toolkit of compiler passes built on top of the kernel. The kernel defines the intermediate representation, provides functions to access and manipulate the intermediate representation, and structures the interface between compiler passes. The SUIF toolkit consists of a set of compiler passes implemented as separate programs. Each pass typically performs a single analysis or transformation and then writes the results out to a file. This is inefficient but flexible. SUIF files always use the same output format so that simply running the programs in a different order can reorder passes. New passes can be freely inserted at any point in a compilation. This approach also simplifies the process of making code modifications and extension.

The paper is organized as follows: Section 2 gives an overview of our ongoing project, and discusses the background of message-passing code generation， the related work as well as some concepts. Section 3 describes our algorithm of automatic communication code generation. Section 4 presents several method of communication optimization. We end our paper with some conclusions in section 5.

2. BACKGROUNDS AND CONCEPTS

2.1. Overview of the KAP project

We use SUIF1.3.5 compiler infrastructure to implement the KAP compiler. It will generate programs in C, which call the MPI library to send and receive data between processors. The algorithms we use to implement our compiler mainly

refer to the work presented by Amarasinghe and Lam [4]. The reason we must reproduce much of that work is because their compiler pass is not an open source compiler, at the same time the quality of the parallel source code generated benefits from many of the features provided by our compiler.

We have implemented in our compiler the construction of linear inequalities system from a loop nest, on condition that the computation decomposition and the exact data-flow analysis are given. This linear inequalities system is then used to generate a loop nest for each read array access, and the loop nest is due to receive and unpack messages from the processor, which generated the value read. The linear inequalities system is also used to generate a loop nest for each write array access, which is to pack and send messages to the processors, which need it.

Our compiler is made up of several passes. First, Fortran/C front end is used to transform the Fortran/C program into the intermediate format file, and at the same time the loop-level parallelism is identified and optimized. Second, data and computation decomposition map the data and computation to the processors, and determine the form of decomposition so that parallelism is maximized while communication is minimized. Third, exact data-flow analysis is used to find the data dependence of every pair of read instance and write instance, and identify accesses to non-local data. Dependence relation information is used to generate send and receive message. The final pass is code generation, which is machine-dependent. It is responsible for carrying out the transformations required by the previous passes and creating the final parallel code. The code generator first schedules the parallel loop nests and arrays so that each processor possesses an array section and executes its allocated iterations, then inserts the necessary send and receive code into the created code.

2.2. Some concepts

1) The linear inequalities system

We use a unified framework based on linear inequalities to handle multi-dimensional integer spaces such as iteration, data and processor spaces. The linear inequalities system is used to generate and optimize the communication code in our compiler.

We represent all possible values of a set of integer variables $(v_1,...,v_n) \in Z^n$ as an n-dimensional discrete Descartes space, where the k-th axis corresponds to the variable v_k. Coordinate $[x_1,...,x_n] \in Z^n$ corresponds to the value $v_1 = x_1,...,v_n = x_n$.

In our compiler algorithms, we use Fourier-Motzkin elimination [7] as one of the key transformations in manipulating systems of linear inequalities.

2) Linear inequalities representations of iteration, data and processor spaces

In our compiler we use linear inequalities system to represent the iteration, data and processor space.

Iteration Space -I

For the n-deep loop nest in Fig.1, where $\vec{v}$ is a symbolic constant vector, l_k and h_k are affine functions. The iteration space **I** is defined as follows:

$$I = \left\{ \vec{i} = (i_1,...,i_n) \in I \middle| \forall k = 1,...,n; \wedge \begin{matrix} i_k \geq l_k(\vec{v}, i_1,...,i_{k-1}) \\ i_k \leq h_k(\vec{v}, i_1,...,i_{k-1}) \end{matrix} \right\}$$

```
for i1=l1(v) to h1(v) do
  for i2=l2(v , i1) to h2(v ,i1) do
    ...
      for in=ln(v , i1...in-1) to hn(v , i1...in-1) do
        ...
                    ...=... A[f(v , i1,...,in)] ...
        A[g(v ,i1,...,in)]=...
```

Fig.1 : Example Loop Nest

Data Space - A

For an m-dimension array A, the data space **A** is defined as follows:

$$\mathrm{A} = \left\{ \vec{a} = (a_1,...,a_m) \in A \mid k = 1,...,m; 0 \leq a_k \leq u_k \right\}.$$

The read and write access functions are affine functions with the form of $f_r(\vec{v}, i_1,...,i_n) = (a_1,...,a_m)$ and $f_w(\vec{v}, i_1,...,i_n) = (a_1,...,a_m)$ respectively, where $\vec{v}$ is a symbolic constant vector, and $(i_1,...,i_n) \in I$, $(a_1,...,a_m) \in A$.

Processor Space - P

For a q-dimension processor array P, the processor pace **P** is defined as follow:

$$P = \left\{ \vec{p} = (p_1,..., p_q) \in P \mid k = 1,..., q; 0 \leq p_k \leq u_k \right\}$$

2.3. Related work

1) Computation and data decomposition

Computation decomposition C, a set of iteration and processor pairs, maps iterations to processors [5]. The computation decomposition mapping n-dimensional iterations to a q-dimensional processor space is a function $\vec{p} = C\vec{i} + \vec{\gamma}$, where C is a $q \times n$ matrix and $\vec{\gamma}$ is a constant vector.

Data decomposition D, a set of array and processor pairs, maps array elements to processors [5]. For each element of an m-dimensional data space, the data decomposition mapping array elements to a q-dimensional processor space is a function $\vec{p} = D\vec{a} + \vec{\delta}$, where D is a $q \times m$ matrix and $\vec{\delta}$ is a constant vector.

When we find the computation and data decomposition of a program, some rules must be followed. First, find whether there is a unified computation and data decomposition in the overall program. If the unified computation and data decomposition do exist, they will be used on every loop nest in the program. If not, find computation and data decomposition as possible as can in the program fragments from large scale. The worst situation is every loop nest has its own computation and data decomposition.

2) Last write tree and data dependence

Data dependence means that there are read and write accesses to the same array element in the loop nest. We use LWT (Last Write Tree) [6] to represent the relation between the read and write instances when we analyze the dependence relation. Every pair of read-write instances corresponds to a LWT. The LWT is a binary tree and represents the exact data-flow information. It is a function that maps an instance of a read operation to the very write instance that produces the value read. We denote a read instance and a write instance by the values of their loop indices, $\vec{i}_r$ and $\vec{i}_s$, respectively. The domain of the LWT function is a set of read iterations $\vec{i}_r$ that satisfy the

constraints imposed by the loop bounds. A LWT has a root node, several leaf nodes and interior nodes. The interior nodes contain the further constraint on the value of the read instance $\vec{i}_r$. Leaf nodes are divided into $\perp$ nodes and non-$\perp$ nodes. With LWT we can create the communication code.

An LWT partitions the iteration set of a loop nest into the contexts of its leaves. If the values read by the iterations in a context $\iota \subseteq I$ are written within the loop, then the context has a last-write relation μ. The last-write relation μ of the context ι is a set of iteration pairs $(\vec{i}_r, \vec{i}_s)$, where $(\vec{i}_r, \vec{i}_s) \in \mu$ iff $\vec{i}_r \in \iota$ and $\vec{i}_s \in I$ is the iteration that generates the value read in iteration $\vec{i}_r$. A context ι can be written as $\{\vec{i} \in I \mid \vec{q}(\vec{v}, \vec{i}) \geq \vec{0}\}$ and a read-write relation μ can be written as $\{(\vec{i}_r, \vec{i}_s) \in I \times I \mid \vec{q}'(\vec{v}, \vec{i}_r, \vec{i}_s) \geq \vec{0}\}$ where $\vec{q}$ and $\vec{q}'$ are vectors of affine expressions.

If iteration $\vec{i}_s$ and $\vec{i}_r$ access the same array element $\vec{a}$, the iteration $\vec{i}_s$ is executed before iteration $\vec{i}_r$, and there has no other iteration modify the element $\vec{a}$, we say that there exists the last write relation between $\vec{i}_s$ and $\vec{i}_r$. The LWT identifies the data dependence relation to the specific array element precisely.

When we analyze the data dependence, we confine the analyzing area in every loop nest in the program. If there is a read access and a write access to the same array, then we will analyze the data dependence in the loop nest, otherwise we will not do.

3) Message Passing Code Generation

The algorithm to generate message-passing code is described in detail in reference [4]. We will only mention briefly how the algorithm works. Suppose we have a loop nest as shown in Fig.1, which contains read and writes accesses to the same array. We can get the computation and data decomposition of the loops in Fig.1. Because there is read and write operation to the same array A, and there may be data dependence. LWT is used to represent the dependence relation[0].

The code generation can be divided into three steps. First, we make use of the data decomposition to create the data distribute code; Second, transform the original loop nest into the execution loop nest with the computation decomposition. Last, insert receives and send loop nest to communicate between the processors. The receive loop nest is to receive and unpack the data from the processor that create the value for each read access to the array location A [f ($\vec{v}$, i1…in)]. The send loop nest is to pack and send the data to the processor that need the value for each write access to the array location A [g ($\vec{v}$, i1…in)].

3. AUTOMATIC COMMUNICATION CODE GENERATION

We define the communication between processors formally as a communication set M, which is a set of elements $(\vec{i}_r, \vec{p}_r, \vec{i}_s, \vec{p}_s, \vec{a}) \in I \times P \times I \times P \times A$, where $(\vec{i}_r, \vec{p}_r, \vec{i}_s, \vec{p}_s, \vec{a}) \in M$ iff processor $\vec{p}_s$ needs to send the value in location $\vec{a}$ in iteration $\vec{i}_s$ to processor $\vec{p}_r$ for use in iteration $\vec{i}_r$. The communication of parallel program is made up of four parts: data distribution, the communication before computation, synchronization communication, and data gathering.

3.1. Data distribution communication

The data distribution is the first step to create the parallel program. The array section is distributed to the processors using the data decomposition. The communication during the procedure is called data distributed communication. Some available compiler such as Paraguin of North Carolina University has no data distribution. The data are simply broadcasted. In our compiler we have implemented the data distribution so that every processor possesses an array section according to the data decomposition. Experiments prove that the performance of the program is improved after using array distribution.

To implement data distribution, first of all we should found a linear inequalities system, which consists of the bound of the array and the data decomposition D. The data decomposition D indicates which dimensions of the array will be distributed. The dimension, which is not to be distributed, will be fully copied to the processors. The algorithm to distribute data is shown in Fig.2.

```
Input:    data decomposition list D_list: D1,...,Dn
Output:   loop nests to send/receive the data
Let:      p: processor

While( D_list!=NULL){
  /* get a data decomposition from D_ list */
  D =D_list → step;
  /* get the array name A from the data decomposition*/
  A ← D;
  /* get the bound of the array*/
  B ← lbi ≤ Ai ≤ ubi
  /*get the data a need to be sent to non 0# processor*/
  a ← (D,B)
  /*create the send/receive code to distribute data*/
  if(p == 0){
    /*create the pack code to pack the data a to be sent*/
    MPI_Pack( )
    /*create send code to send data to non 0# processor */
    MPI_Send( )
    }
  Else{
   /*create receive code to receive data from 0# processor*/
    MPI_Recv( )
    /*create unpack code to release data */
    MPI_Unpack( )
    }
}
```

Fig. 2: The Algorithm to Distribute Ddata

3.2 The communication before computation

The communication before computation occurred before executing any computation of the loop nest. The communication before computation is needed when the data decomposition and computation decomposition are not aligned completely. With the context of $\perp$-node in LWT the compiler can simply load all the non-local data onto a processor before executing any of the computation. Given computation decomposition and an initial data decomposition produced by an earlier compiler phase, the technique to generate the necessary communication code is

no different from that used in the location-centric approach. With data dependence provided by the context of $\perp$-node in LWT, data and computation decomposition, and the bound of the original loop nest, a linear inequalities system can easily be established. Then, we use the FME to create the communication code to send or receive data between processors based on the linear inequalities system. The algorithm to create the communication code before computation is shown in Fig.3.

3.3. Synchronization communication

Input: *data decomposition matrix: D*
computation decomposition matrix: C
read access function list f_r_list: $\vec{f}_{r1}, \vec{f}_{r2}, ..., \vec{f}_{rm}$
data dependence: ι
Output: *loop nests to send/receive the data*
Let: *p: processor*
P_r: receive processor,
P_s: send processor

For each p
$P_s \leftarrow p$
While(f_r_list!=NULL) {
fr=f_r_list $\rightarrow$ step
*/*get array name from f_r , represent as A*/*
$A \leftarrow f_r$
*/*use relation* $(\vec{a}, \vec{p}_s) \in D$ *to get array elements of P_s */*
$a \leftarrow (D_a, P_s)$
*/*use relation* $\vec{a} = \vec{f}_r(\vec{v}, \vec{i}_r)$ *to get the read Iteration I_r*/*
$I_r \leftarrow (a, f_r)$,
*/*use the relation* $(\vec{i}_r, \vec{p}_r) \in C$ *to get P_r */*
$P_r \leftarrow (I_r, C) \cap \vec{i}_r \in \iota$
If(P_s!=P_r) {
*/*create pack code, the data sent is a*/*
create MPI_Pack()
*/*create send code to send data to p_s*/*
create MPI_Send()
}
}
$P_r \leftarrow p$
While(f_r_list!=NULL) {
f_r=f_r_list $\rightarrow$ step
*/*get array name from f_r , represent as A*/*
$A \leftarrow f_r$
*/*use the relation* $(\vec{i}_r, \vec{p}_r) \in C$ *to get the Iteration P_r has*/*
$I_r \leftarrow (\mathrm{Pr}, \mathrm{C}) \cap I_r \in \iota$
*/*use the relation* $\vec{A} = \vec{f}_r(\vec{v}, \vec{i}_r)$ *to get the data a */*
$a \leftarrow (I_r, f_r)$
*/*use relation* $(\vec{a}, \vec{p}_s) \in D$ *to get array elements of P_s */*
$P_s \leftarrow (D, a)$
If(P_s!=P_r) {
*/*create receive code to receive the data sent from p_s*/*
create MPI_Recv()
*/*create unpack code to release the data*/*
create MPI_Unpack()
}
}

Fig.3: Algorithm to Create the Communication Code before Computation

Since the iteration of reading and writing to the same data is executed on different processors, the synchronization communication must be applied during the computations. To get the data read which is written on other processor, we need to communicate with the processor, which possesses the data. With data dependence provided by the context of non-$\perp$-node in LWT, data and computation decomposition, and the bound of the original loop nest, a linear inequalities system can be established. And then the synchronization communication code can be generated from that linear inequalities system. Communication and computation are more tight-coupled for the non-$\perp$-nodes. The LWT specifies all the pairs of iterations that share a producer and consumer relationship. By applying the computation decomposition function on the related iterations, we can derive the identity of the processors that read and write the same value. If the writer and reader are different processors, then communication is necessary. The algorithm to create the synchronic communication code is shown in Fig.4.

Input: *data decomposition matrix: D*
computation decomposition matrix: C
exact data dependence relation: μ
read access function: f_r
write access function: f_s
Output: *loop nests to send/receive the data*
Let: *p: processor*
P_r :receive processor
P_s :send processor

For each p
$P_s \leftarrow p$
*/*use relation* $(\vec{i}_s, \vec{p}_s) \in C$ *to get write Iteration I_s*/*
$I_s \leftarrow (C, P_s)$
*/*use relation* μ *to get the read Iteration I_r*/*
$I_r \leftarrow (I_s, I_r) \in \mu$
*/*use relation* $(\vec{i}_r, \vec{p}_r) \in C$ *to get P_r */*
$P_r \leftarrow (I_r, C)$
If (P_s != P_r){
*/*use relation* $\vec{A}_s = \vec{f}_s(\vec{v}, \vec{i}_s)$ *to get read data A_s*/*
$A_s \leftarrow (I_s, f_s)$
*/*pack the data A_s need to be sent */*
create MPI_Pack( )
*/*create send code from P_s to P_r */*
create MPI_Send( )
}
$P_r \leftarrow p$
*/*use relation* $(\vec{i}_r, \vec{p}_r) \in C$ *to get I_r*/*
$I_r \leftarrow (P_r, C)$
*/*use relation* μ *to get write Iteration I_s */*
$I_s \leftarrow (I_s, I_r) \in \mu$
*/*use relation* $(\vec{i}_s, \vec{p}_s) \in C$ *to get p_s */*
$P_s \leftarrow (I_s, C)$
If (P_s != P_r) {
*/*use relation* $\vec{A}_r = \vec{f}_r(\vec{v}, \vec{i}_r)$ *to get read data A_r*/*
$A_r \leftarrow (I_r, f_r)$
*/*create receive code to receive the data from P_s */*
create MPI_Recv( )
*/*create unpack code to release the data*/*
create MPI_Unpack( )
}

Fig. 4: Algorithm to Create the Synchronization

3.4. Data gathering

The communication of data gathering occurred after all computations. The goal is to gather the computation result of the loop nest on the processors to the main processor. The procedure is opposite to the data distribution; the algorithm is omitted in this paper.

4. OPTIMIZATION OF COMMUNICATION

The algorithms given in Section 3 are used to create the communication code in automatic parallelization code generation for distributed memory machines. The code created needs optimization because it may be inefficient.

There are several methods to optimize the communication such as overlapping communication with computation, message aggregation, and message coalescing and so on.

4.1. Overlapping communication with computation

Overlapping communication with computation is implemented by merging loop nests of computation, send and receive codes for each communication set [4]. A processor checks iteration if it belongs to the computation domain, and if it is to take part in each of the communication sets. If the condition is satisfied, the computation, send and receive loop can be merged.

4.2. Message coalescing

Separate communication for different references to the same data is unnecessary if the data has not been modified between uses. When statically analyzing the access patterns, these redundant communication operations are detected and coalesced into a single message, allowing the data to be reused rather than communicated for every reference. For sections of arrays, which are not disjoint, unions of their overlapping communication descriptors ensure that each unmodified data element is communicated only once. Such software-based caching is always beneficial since entire communication operations can be eliminated. Message coalescing [8] is divided into two kinds: dynamic message coalescing and static message coalescing. The former is also called message caching.

4.3. Message aggregation

Multiple messages (corresponding to several array sections) to be communicated between the same source and destination can also be aggregated into a single larger message. Message aggregation [9] can be divided into two kinds: (1) self aggregation where messages generated by different instances of the same access are aggregated, (2) group aggregation where messages generated by different accesses are aggregated.

4.4. Message vectorization

Non-local elements of an array that are indexed within a loop nest can also be vectorized into a single larger message instead of being communicated individually. This optimization method is called Message vectorization [10]. The “itemwise” messages are combined, or vectorized, as they are lifted out of the enclosing loop nests to a selected level. Vectorization reduces the total number of communication operations, but at the cost of increasing the message length. For this reason, vectorization should perform well on machines with high communication overheads.

5. CONCLUSIONS

[0]In this paper, we introduce several algorithms to generate communication code of parallel program and present some methods of communication optimization. However, the communication code may be inefficient. So the code created needs to be optimized. That is our next target.

6. REFERENCES

[1] B. William, D. Ramon, et al, “parallel programming with Polaris,” Computer, 1996, 29(12): 78-82.

[2] R. P. Wilson, R. S. French, et al, “SUIF: An infrastructure for research on parallelizing and optimizing compilers”, ACM SIGPLAN Notices, December 1994, 29(12): 31-37.

[3] C.Q. Zhu, B.Y. Zang, et al, “An automatic parallelizer”, Journal of Software, 1996, 7(3): 180-186.

[4] S. P. Amarasinghe, M. S. Lam, “Communication optimization and Code Generation for distributed memory machines”, Proc. of The ACM SIGPLAN '93 Conference on Programming Language Design and Implementation (PLDI), Albuquerque, June 1993, 126-138.

[5] J. M. Anderson, M. S. Lam, “Global Optimizations Parallelism and Locality on Scalable Parallel Machines”, Proceedings of the SIGPLAN ‘93 Conference on Program Language Design and Implementation, June 1993.

[6] D. E. Maydan, S. P. Amarasinghe, et al, “Array data-flow analysis and its use in array privatization,” Proceedings of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Charleston, South Carolina, Jan. 1993, 2-15.

[7] C. Ancourt, F. Irigoin, “Scanning polyhedra with DO loops”, Proceedings of third ACM SIGPLAN Symposium on Principles & Practice of Programming Languages (PPOPP), Williamsburg, Virginia, Apr. 1991, 39-50.

[8] A. Kubota, I. Miyoshi, et al, “TINPA: A parallelizing compiler for message-passing multiprocessors”, Proc. Of int’l Symposium on Parallel and Distributed Supercomputing, September 1995, 214-223.

[9] C.D. Pham, Albrecht, “Optimizing Message Aggregation for Parallel Simulation on High Performance Cluster”, Proceedings of MASCOTS’99.

[10] E. Su, A. Lain, et al, “Advanced compilation techniques in the PARADIGM compiler for distributed memory multicomputers”, Proceedings of the 9th ACM International Conference on Supercomputing, Barcelona, Spain, July 1995, 424-43.

The Means of Flowgraph Stream Parallel Programming for Clusters*

V.P. Kutepov1, V.A. Lazutkin2, Liang Liu3, M.A. Osipov4
Chair of Applied mathematics
Moscow Power Engineering Institute (Technical University)
ul. Krasnokazarmennaya 14, Moscow, 111250 Russia
1Email: kutepovvp@mpei.ru
2Email: vilazag@yandex.ru
3Email: liuliang_pmo@yahoo.com.cn
4Email: osipov@rbc.ru

ABSTRACT

Flowgraph stream parallel programming software for clusters is described in the paper. The software includes a module language of visual parallel programming, tools for programs development and their performance on clusters.

Keywords: FSPPL, FGPP, Parallel programming, Cluster.

1. INTRODUCTION

A widening production and use of high performance computing systems, in particular clusters, involve researches in the development new parallel programming languages and tools providing efficient writing, debugging, optimizing and running parallel programs.

The most parallel programming languages are problem oriented and they are developed by extending sequential languages. A good example is High Performance Fortran (HPF), which is based on Fortran 90 and directed to parallel programming computing tasks. HPF is a set of extensions of Fortran 90 standard that permits the programmer to specify how data is to be distributed across multiple processors. HPF's constructs allow the programmer to indicate potential parallelism at a relatively high level, without entering into the low-level details of message-passing and synchronization. Special instructions are introduced in HPF in order to programmer can explicitly distribute data over processors of computing system and parallelize computations taking into account number of processors.
The same approach to explicit parallelization of computation process by providing special instructions or message passing constructs in order to synchronize parallel process is used in many extensions of functional, logical, object oriented and other languages.

MPI (Message Passing Interface), PVM (Parallel Virtual Machine) can be considered as standards for programming interactions between parallel processes, however the development of parallel programs by using these means is a tedious complicated task.

The next step in widening of means for synchronization and interactions of the processes is multithreading, which is popular now due to multicore computers. On our opinion multithreading is well known as low level means for representation in the program interactions between processes, though possibilities to parallelize processes with multithreading are richer than by using MPI, resulted complications in programming process is no less.

* This project is supported by the Russian Foundation for Basic Research (No. 06-01-00817).

It is necessary to note that there are functional and logical languages with implicit representation of parallelism in a program and explicit parallel operational semantics. Flowgraph stream parallel programming language described in present paper is like the module. It is based on clear structural visual forms of the program representation and using conventional languages for module programming.

Spatial parallelism induced by data independent modules and pipeline (data flow) parallelism are equally and easily can be expressed in a program. Moreover, fine grained parallelism can be used in module programs. The realization of the language on clusters comprise three main components – language, tools supporting programming process and control system managing parallel computations, which have been integrated provide efficiency of the parallel computations.

2. THE FLOWGRAPH STREAM PARALLEL PROGRAMMING LANGUAGE (FSPPL)

The FSPPL is directed toward large-grained stream programming and can also be applied efficiently for program modeling of distributed systems, queuing systems, and others, with interaction between their components being structured and controlled by the data flow [2].
The language allows the following three types of parallelism to be efficiently and uniquely represented in program:
– the parallelism of data-independent fragments;
– the flow parallelism, conditioned by pipeline data processing;
– the dataset parallelism (i.e. SIMD parallelism) implemented in the FSPPL through the tagging mechanism, when one and the same program or its fragment are applied to different data.

Other important (from the programming viewpoint) features of the FSPPL are the following:
– the possibility of a visual graphical and textual representation of programs;
– the possibility of a simple strategy structuring of programs and reflecting the decomposition hierarchy, which is based on the graph-subgraph relationship;
– the use of conventional sequential languages for module programming.

2.1 The graphical version of FSPPL

A flowgraph parallel program (FGPP) is represented as a pair of the form $<FG, I>$, where FG is the flowgraph, and I is its interpretation.

The flowgraph or simply graph makes it possible to visualize the structure of program, which is being built on

the basis of modules. The interpretation associates with each module a set of subroutines and with inputs and outputs of modules – types of the data, which are passed between subroutines of modules during the execution of FGPP. All its inputs and outputs of module are strictly typified and divided into groups (paying a tribute to the prehistory of the FSPPL, which are called conjunctive input groups (CIG) and conjunctive output groups (COG)), reflecting the structure of data, which flows between module subroutines. Each CIG of the module is assigned to a subroutine written in a conventional programming language (C/C++, Pascal, Java, etc.), and the types of formal parameters in the subroutine must respectively coincide with the display order (from left to right) of CIG inputs and their types. In addition, an integer variable tag is added to the subroutine as a formal parameter specified to be the first in the list of parameters. This parameter is not represented on FG, because it serves as an identifier for data, which is moved between the FGPP modules during their execution of subprogram.

The graphical display of a module, which is necessarily accompanied with its name, also can carry supplemental information that is presented in the form of comments disclosing its function, the function of subroutines specified by CIGs, etc. The representation of module is given in Fig. 1..

The FG has a modular structure, connecting the COG outputs of a module with CIG inputs of the same or another module. The connected inputs and outputs of modules must be of the same type.

Some inputs of module can be free of connections (free inputs) and data values should be assigned to them before execution of FGPP.

A CIG of module is also allowed to be free of inputs. In this case, the subroutine corresponding to this CIG has no parameters and starts to be carried out together with the beginning of FGPP execution. It operates as a generator of data retrieved from a carrier or produced by the subroutine separately.

In an operational semantic of FSPPL with links between modules of the FGPP are associated unique buffers storing data asynchronously transferred to the CIG module inputs (the actual parameters of corresponding subroutines), which are produced by the subroutines of predecessor-modules and directed to the COG module outputs. In parallel processing of the FGPP, the data-tagging (assigning a tag to data) mechanism allows for the uniqueness of the relationship between the result and input data.

An important element of the FGPP structure is the concept of a subgraph, which makes it possible for a parallel program to be structured efficiently and a FG to be represented as a set of a few levels reflecting the decomposition stages of the initial problem in constructing the parallel program. Graphically, the subgraph is built similarly to the FG and, in its content, is a separate FGPP fragment inserted instead of one or more modules of a higher level subgraph in the graph–subgraph hierarchy (see Fig. 1.).

The visual graphical design of the FGPP is provided through special-purpose programming tools (see Section 4).

2.2 The symbolic representation of FGPP

In symbolic representation, the FGPP is a set of XML-files that hierarchically describe the FG structure and interpretation. The choice of XML is explained by the fact that, first of all, it is a commonly accepted standard for the description of hierarchy constructs (and many existing methods have been developed to deal with it), and secondly, the internal representation of the FGPP is also stored in the XML format. So, it will be most reasonable also to have the symbolic representation in the XML format.

The description of the FG structure is located in a separate XML-file, which contains all FG-modules and links between them, and each module also has its own (already separate) XML-file with a detailed description, including all about its CIGs, COGs, and the interpretation of CIGs (i.e., the subroutines assigned to each CIG). If the FG includes subgraphs, their XML-representation is constructed similarly to the XML-description of the FG.

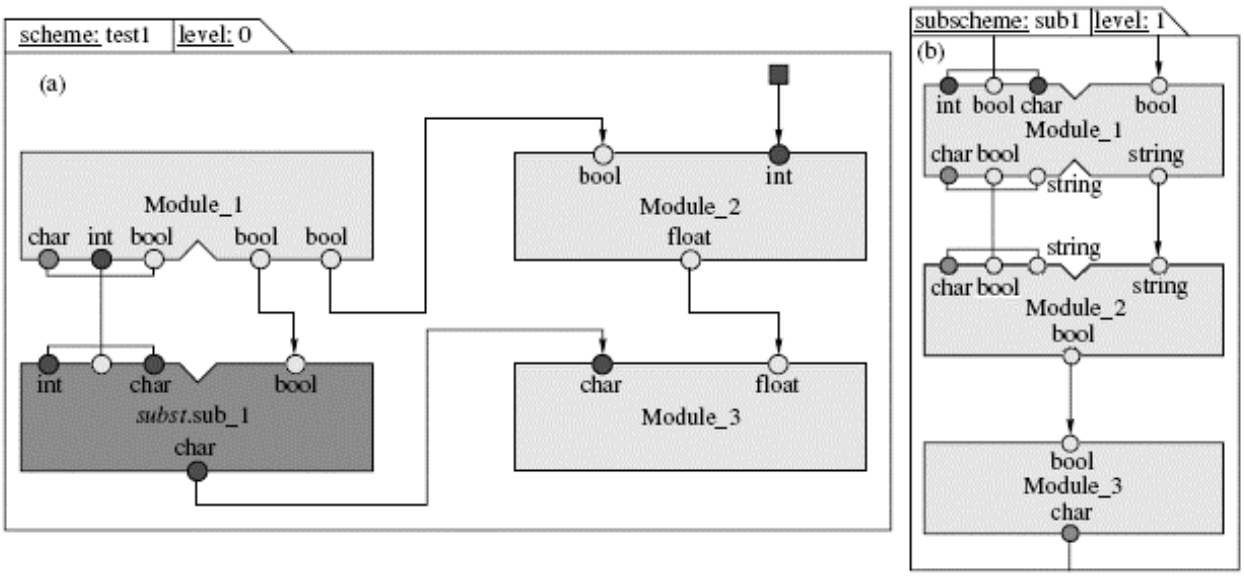


Fig. 1. Structure of FG

2.3 The operational parallel semantics of FSPPL

The parallel execution of FGPP is represented as a sequence of alternating states, each of which is characterized by a set

of processes induced during the execution of the FGPP-module subroutines.

In data flow model of computation, it is supposed that each data-processing unit (a fragment of the program) can be run, if there are data on its input, and in addition the data processing unit cannot be initiated once more (though there are data in input queue) till proceeding computation terminates. Thus, in any state, the set of FGPP processes, following this stream execution principle, is no more than the total number of CIGs of all modules in FGPP.

In model of parallel execution of FGPP, three types of parallelism are realized:

– spatial parallelism, which reflects the data independence of the FGPP modules, and, consequently, the execution of subroutines assigned to their CIGs;
– the data flow parallelism mentioned above;
– SIMD-parallelism, which supposes simultaneous application (and execution) of many copies of the same subroutines to data at inputs of respective CIG marked different tags on.

The uniqueness of the relation between input and output values in the parallel execution of FGPP can be provided through a tagging mechanism. In the operational semantics of the FGPP language, this means that, if at all inputs (in input buffers) of CIG there are data marked by corresponding tags, one runs a few processes in parallel, each of which is uniquely identified by a tag and the data assigned to it. When the results are transferred from one module to another, the tags are inherited to allow the history of the processing of different data to be distinguished.

Now, we describe the process of the FGPP execution.

(a) The FGPP module is assumed to be ready for running by any of its CIGs, if in all its inputs (in the corresponding buffers) there are data marked by the same tag.

Different tags identify different data to which respective to the CIG of copies of subroutines can be applied simultaneously, realizing SIMD parallelism. An order of the execution these processes does not violate unique of computing process due to tagging mechanism.

The modules with no-input CIGs (which correspond to subroutines with an empty set of parameters) are assumed to be ready for execution by these CIGs from the time of the FGPP execution initialization; however, the processes induced by them can be generated only once.

(b) In executing of a process, a subroutine can use the special-purpose statements (implemented by a common call to special functions): WRITE and READ, which provide an interface between modules, i.e. build various schemes of data exchange between subroutines of different modules through reading data from or writing data to the buffers, assigned to the module CIG inputs.

When WRITE is executed, the context of the subroutine (more exactly of the process) that initialized the statement is retained, and after its execution, the process keeps operating in the interrupted context (a common mechanism of return after a subroutine call).

When READ is executed, the context of the process is also retained, and after its termination, the process keeps operating in the previous context. The READ statement allows the process to read data with the indicated tag from buffers assigned to the CIG, which initiated the process. The data with the indicated tag retrieved from the listed CIG inputs are assigned to variables in the list of variables of the READ statement (the values returned to the process). In the execution of the READ statement, if the requested data have not yet arrived to the buffer memory, the execution is delayed until the data arrive. The arrival time is controlled for any recording of data into the buffer memory of the corresponding CIG. After the READ statement execution, the requested data are deleted from the buffer, and the subroutine context induced the READ statement is recovered. The READ statement makes it possible to arrange a flow-dependent operation mode for the running process, sequentially reading and processing the data that arrived at the CIG inputs.

Note that the semantics of the FGPP (and its implementation) implies that, for storing data arriving through links between modules, and the buffers are assigned (arranged during the execution) to CIGs rather than COGs of modules, since each link originating in a COG of any module goes necessarily to the input of a CIG of some FGPP module.

For a more sophisticated operation with data arriving to module CIGs (in particular, with the assigned buffers), the statement CHECK (<number of CIG>, <tag>, <list of inputs>, <variable>) is provided, which checks the availability of data with the indicated tags at the CIG inputs. This statement allows the process to make an independent decision on its actions depending on the data availability. As a result of this statement, a variable is assigned with the total number of data with the specified tag at the CIG inputs, whose numbers are given in the list of inputs.

For learning with which tag when this or other subroutine has been started, the GETTAG() statement is used. After its execution the GETTAG() statement retracts the tag value of data set, with which this subroutine is executed.

3. THE DESIGNING OF FLOWGRAPH STREAM PARALLEL PROGRAM

FSPPL is directed to the parallelization of large scale tasks and it very well does for programming queuing systems, modeling distributed systems, in particular controlling and computing systems, assembling complicated applications with various modules, etc.

A process of programming in FSPPL is complicated problem and it has a number of stages:

– decomposition of the problem into subproblems in a such a way that behind of each of them stands an image of a module or subscheme; the decomposition process is iterative by its nature (well-known in programming top-down strategy of designing a program);
– associating with each modules strictly defined set of functions, which module should be able to perform and fixing CIGs, their inputs and types for every module function;
– elaborating data flows between modules, structuring them by introducing modules COGs and connecting outputs of COGs with respective CIGs of modules in FG.

Resulted FG of parallel program will be developed after some repeating steps of these stages.

It is supposed that there are one source module in FG (this module has no inputs usually) and one module without outputs symbolizing final state of an execution of FSPP.

The FG can be considered as structural representation of a problem solution and as graph scheme of the parallel program in FSPPL.

In realization FSPPL all stages of constructing FG are automated and visualized and supported with comprehensive programming tools (see Section 4).

Being FG developed the next step is programming functions of modules as some kind of subroutines. Any suitable sequential programming language can be used for

this purpose, moreover, for programming different functions can be applied different languages. The only problem which is arising with using different languages is coordination of different formats of the same data types in used languages.

In spite of seeming simplicity of the subroutines programming programmer should provide uniqueness of the correspondence between input data and result in data flow parallel execution of the program. The tagging mechanism in FSPPL provides solution of this problem by inheriting tag with all data produced individual computation process.

In order to do more parallel program WRITE instructions in subroutines should be performed as earlier in execution process of subroutine as it possible.

It is often possible to transform spatial parallelism into data flow (stream like) parallelism and vise versa. Though this cannot change degree of parallelism in principle however spatial form is often preferable due to decentralizing controlling functions during program execution process. A number of computers in cluster can be profitably take into account in transformation process.

Instructions READ and CHECK in subroutine allow programmer to realize sophisticated data stream dependent computations, for example, computations with sequence of data ingoing asynchronously at inputs of a CIG of a module.

Our experience with programming different kind of complicated tasks shows that programming process in FSPPL, supporting by developed tools (see Section 4), permits to avoid many errors in program project solutions and speed up developing parallel program.

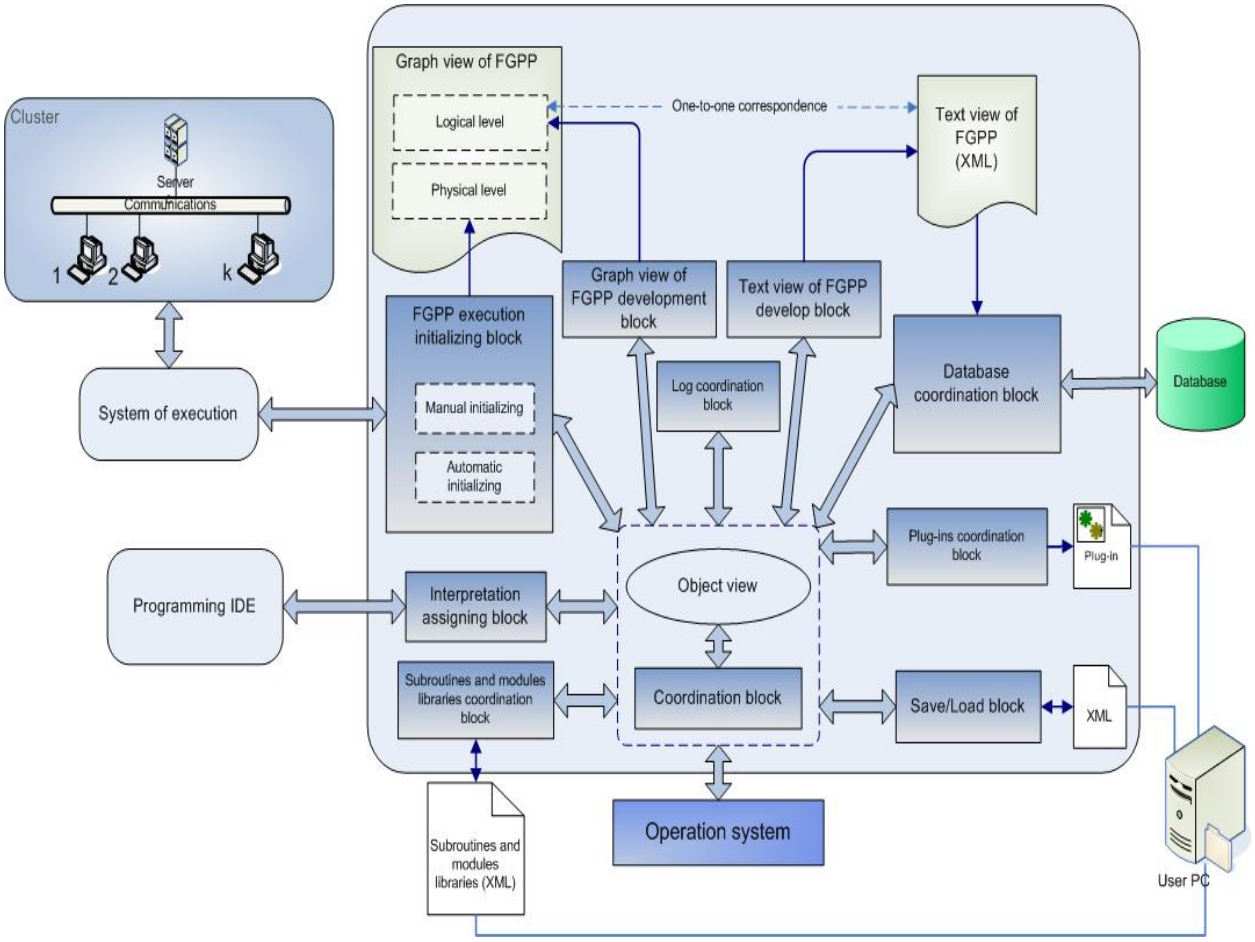


Fig. 2. Architecture of programming environment

4. ENVIRONMENT OF PROGRAM DESIGN IN FSPPL

The purpose of the environment – automation of programming process in FSPPL. Tools of this environment provide:

– visual graphic development of FGPP;
– textual development of FGPP;
– development of module subroutines.

The architecture of the programming environment is given in Fig. 2..

The coordination block – provide management of all blocks of the environment and the coordination of information flows.

The object view of FGPP – the object-oriented internal structure of environment. Together with the coordination block, the object view forms a valuable managing system, which realize an interaction between blocks of the environment and the operational system.

The graph view of FGPP is divided into two levels: logical and physical. In the logical level the graph view provides visual presentation of FGPP. The physical level displays mapping modules of FGPP to computers of cluster.

The text view of FGPP is the completely consistent view of the program with the text version of FSPPL.

The graph view of FGPP development block is a toolkit for visual construction of FGPP. It carries out next

functions:
– addition, modification and deletion of modules and
– realization of connections between modules;
– displaying and sealing any fragments of FG and etc.

The text view of FGPP development block is designed for the program construction, in the form of XML-files, which format is completely coordinated with the text version of FSPPL.

The Interpretation assigning block contains toolkit, by means of which the user chooses a language for programming subroutines of the modules.

Subroutines and modules libraries coordination block allows user to organize the libraries of the subroutine files.
Programmers can also use Database coordination block for long term storing developed parallel programs, histories of program designing process and any valuable information associated with programming activity.

The FGPP execution initializing block is intended for mapping modules of FGPP into computers of the cluster that can be considered by execution system as preliminary manual step of planning parallel execution of the program on cluster.

Programmers can receive with this block visual picture of initial allocation modules to computers of the cluster. In fact, manual cutting of FG into subgraphs (in such a way that a balance of computers load of cluster can be provided) proceeds of the allocation procedure.

5. OPERATIONAL MEANS FOR PROGRAM PERFORMANCE ON CLUSTER

An efficiency of parallel computations is formulated as a problem of minimization of both the execution time of a parallel program and the used resources (normally, the number of computers or processor units in the computing system). This problem is actually solved through the scheduling mechanisms and requires a fine account for the features of the running parallel program, the degree of its parallelism, the load of different units of the computing system (computers, communications, etc.), failure s and recoveries, etc [3, 4, 5].

This problem requires a special study; therefore, we will mostly consider the designing aspects of the control of FGPP parallel execution on clusters.
In the structure of this control, one can separate the following two relatively independent levels:
– the general-system level, which controls the cluster configuration and is responsible for the loading control and planning (of nodes or computers) and response to a failure of the cluster and other components;
– the level of proper control of parallel processes induced during the FGPP execution.

The Fig. 3. demonstrates the structure and main blocks of the control of the FGPP execution on cluster systems. Let us describe briefly the functions of these blocks.
AD is the administration block, which installs control software on the cluster, receives data on the cluster operation (for example, loading), and can be intervened in by the administrator, if necessary.

CC is the configuration control block, which configures the cluster nodes (or the whole cluster), reconfigures it in the course of operation conditioned by failures and recoveries of the components (computers, communications, etc.) as well as by the need of dynamic scaling.

IN is the block of FGPP initialization.

FT is the fault-tolerance block, which reacts to the failures of cluster components and implements the accepted strategy of periodic preservation of the states of cluster units for the purpose of a response in the case of their failure. The strategy can be a simple one, which periodically preserves the state of a computer or other controlled components in the shared memory (for example, in the disk memory of the node server) or a more complex one, which, for example, is based on a certain type of "arrangement" between computers on the periodic preservation of their own states in the case of failure [5]. The cluster-component failures or recoveries that are fixed by the FT block are also passed to the CC block as data for changing the cluster configuration.

EX is the block of interfaces (data and message exchanges) between cluster computers in line with an agreed protocol.
LC is the block of the cluster load control and prediction of its change, as well as the redistribution of parallel processes between node computers or cluster nodes, aimed at achieving a peak efficiency of the cluster operation. This block is closely connected with the block of process planning (SC) of each computer, which transfers all necessary data upon computer loading.

The coordination block (CO) is a monitor in the program sense and provides all interfaces between the mentioned blocks.

The process-level blocks are directly related to the management of readiness and identification of the processes induced during the FGPP execution (the buffer control block (BC)), control of the processes states (the process control block (PC)), and the execution scheduling (the scheduling block (SC)). Through calls to the operation system, the PC block can queue the process and identify the statements, which demand calls to the BC block matching the FGPP CIGs (READ, WRITE, OUT, and CHECK).
The blocks implementing these actions constitute the interpreter of FSPPL (see Fig. 3.).

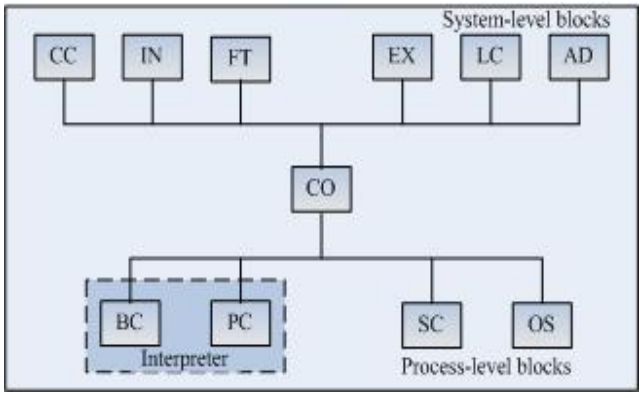


Fig. 3. Structure and main control blocks of the FGPP execution

6. CONCLUSIONS

At present, the development of the tool design environment has been completed, and a test version of the program tools has been created for parallel performance on cluster systems of FGPP.

This test version of the control system was implemented in JAVA. The choice of this programming language is primarily explained by the requirement of support for different OS. This language has the benefit of supporting

parallel programming and synchronization tools, and offering a simple documentation of the system. A deficiency of the language is that JAVA programs are slower than the programs written on such programming languages as C/C++. It is expected to rewrite the critical components or the whole system in "faster" programming languages for a few widespread OS (particularly, for Win32 and Unix/Linux).

Note that the described tools for stream computations implemented in the FSPPL have already been used successfully in the development of software for distributed systems: flexible computer-aided manufacturing systems [1], control systems for military operations, etc. It seems likely that they can be competitive for distributive computations represented as object-oriented programs. To do this, it will suffice to compare the UML language and FSPPL in the context of available tools for the description of parallel and distributed data processing.

7. REFERENCES

[1] A. A. Tikhonov, Extended Abstract of Candidate's Dissertation in Technical Science (in Russian), Moscow: Moscow Power Engineering Institute, 1989.

[2] D.V. Kotlyarov, V.P. Kutepov, et al, "Flowgraph Stream Parallel Programming and Its Implementation on Cluster Systems", Journal of Computer and Systems Sciences Internal, 2005, 44(1): 70-89.

[3] V. P. Kutepov, Arrangement of Parallel Computations in Systems (in Russian), Moscow, Moscow Power Engineering Institute, 1988.

[4] V. P. Kutepov, "On Intelligent Computers and Large-Scale Computer Systems of a New Generation", Journal of Computer and Systems Sciences Internal, 2003, 35(5).

[5] V. P. Lobanov, Extended Abstract of Candidate's Dissertation in Technical Science (in Russian), Moscow, Moscow Power Engineering Institute, 1985.

A New Method of Solving Nash Equilibrium Problems

Wenbo Xu, Min Yu
Institute of Information Technology, Southern Yangtze University, Wuxi, 214122, China
E-MAIL: yuminfeier@163.com

ABSTRACT

Quantum-behaved particle swarm optimization （QPSO） algorithm has been developing rapidly and has been applied widely since it was introduced, as it is easily understood and realized. Nash equilibrium is one of the advanced analysis method in modern economics. In this paper, the Nash equilibrium solution is discussed and given by using QPSO. The effectiveness of the algorithm is proved by experiments.

Keywords: Quantum-behaved particle swarm optimization （QPSO）algorithm, Nash equilibrium, stretching technique, game.

1. INTRODUCTION

Since the 1950s economists have applied game-theoretical concepts to a wide variety of economic problems. The Nash equilibrium concept has proven to be a powerful instrument in analyzing the outcome of economic processes. Since the late 1980s economists have also shown a growing interest in the application of evolutionary game theory. Nash equilibrium as the possible outcomes of the game, evolutionary game theory teaches us to explicitly model the behavior of individuals outside equilibrium. This may provide us with a better understanding of the dynamic forces within a society of interacting individuals. Particle swarm optimization (PSO) algorithm originally introduced by Kennedy and Aberrant in 1995,is a population-based evolutionary computation technique. The ideas that underlie PSO are inspired not by the evolutionary mechanisms encountered in natural selection, but rather by the social behavior of flocking organisms, such as swarms of birds and fish schools. Because Franks Van den Bergh had already proven the PSO algorithm couldn't restrain with the global minimizes, even the local minimizes, and many scholars and many methods improve the algorithm to restrain the performance. In 2004, sun etc introduce quantum theory into PSO and propose a Quantum-behaved PSO based on Delta potential well (QPSO) algorithm. The experiment results testified that QPSO works better than other algorithm.

2.QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION

2.1 Particle swarm optimization

Particle swarm optimization (PSO) is one of the evolutionary computation techniques. The PSO algorithm is described as below [3]:

$$v_{id}^{k+1} = \varpi \times v_{id}^{k} + c_1 rand() \times (p_{id} - x_{id}^{k}) + c_2 \times rand() \times (p_{gd} - x_{id}^{k}) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (2)$$

Where c_1 and c_2 are position constant, and rand()is a random functions in the range [0,1]; $x_i = (x_1, x_2, \ldots x_N)$ represents the *i*th particle ; $p_i = (p_1, p_2, \ldots p_N)$represents the best previous position of the *i*th particle, the symbol g represents the index of the best particle among all the particle in the population; $v_i = (v_1, v_2, \ldots, v_N)$ represents the rate of the position change (velocity)for particle i.

2.2 Quantum-behaved particle swarm optimization

In 2004, sun etc introduce quantum theory into PSO and propose a Quantum-behaved PSO based on Delta potential well (QPSO) algorithm. In Quantum-behaved Particle Swarm Optimization (QPSO), the particle moves according to the following equation [1,2]:

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \ldots, \frac{1}{M}\sum_{i=1}^{M} P_{id}\right) \quad (3)$$

$$p_{id} = \varphi * P_{id} + (1-\varphi) * P_{gd} \qquad \varphi = rand \quad (4)$$

$$x_{id} = p_{id} \pm \alpha * |mbest_d - x_{id}| * ln(\frac{1}{u}) \quad u = rand \quad (5)$$

Where $mbest$ is the mean best position among the particles, p_{id}, a stochastic point between P_{id} and P_{gd}, is the local attractor on the d th dimension of the i th particle, φ is a random umber distributed uniformly on [0,1], μ is another uniformly-distributed random number on [0,1] and α is a parameter of QPSO that is called Contraction-Expansion Coefficient. The QPSO Algorithm is described as follows.

1) Initialize an array of particles with random position and velocities inside the problem space.
2) Determine the mean best position among the particles by (3).
3) Evaluate the desired objective function for each particle and compare with the particle's previous best values.
4) Determine the current global position minimum among the particle's best position.
5) Compare the current global position to the previous global.
6) For each dimension of the particle ,get a stochastic point between P_{id} and P_{gd} by stochastic equation（4）.
7)Attain the new position by stochastic equation（5）.
8) Repeat steps 2)-7) until a stop criterion is satisfied or a pre-specified number of iterations are completed.

3. STRETCHING TECHNIQUE

A different recently proposed technique, developed to address the problem of local minima, is the stretching technique [3]. This technique consists of a two-phase transformation of the objective function. In order to alleviate this problem the following two-stage transformation in the form of the original function f(x) can be applied soon after a local minimum x' of the function *f* has been detected:

$$G(x)=f(x)+\gamma_1\|x-x'\|(sign(f(x)-f(x'))+1) \quad (6)$$

$$H(x)=G(x)+\gamma_2\frac{sign(f(x)-f(x'))+1}{\tanh(\mu(G(x)-G(x')))} \quad (7)$$

where γ_1 , γ_2 and μ are arbitrary chosen positive constants, and sign(.) Defines the well-known three valued sign function.

$$sign\ (x)=\begin{cases}1 & x>0\\ 0 & x=0\\ -1 & x<0\end{cases}$$

The Algorithm is described as follows:

PROCEDURE SQPSO
Begin
t=0
Initialize an array of particles with random position
While (until a stop criterion is satisfied)
Begin
Apply QPSO to f(x)
If find x',
Then
f(x)=H(x)
t=t+1
End while
End begin

The problem considered is a notorious two-dimensional test function, called the Levy No.5:

$$f_6(X)=\sum_{i=1}^{5}[i\cos((i-1)x_1+i)]\sum_{j=1}^{5}[j\cos((j+1)x_2+j)]+(x_1+1.42513)^2+(x_2+0.80032)^2$$

consider $\gamma_1=10000$, $\gamma_2=1$, $\mu=10^{-10}$, $c_1=c_2=0.5$. the swarm of size 20，initialized into the cube [-2，2].After run 100 using QPSO and stretching technique, the result as following:

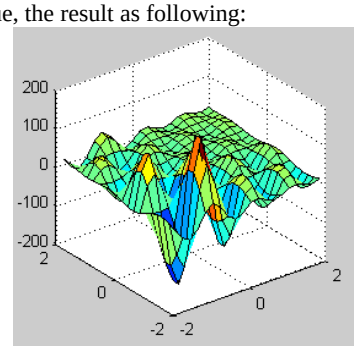


Fig 1 The original plot of the function Levy No.5

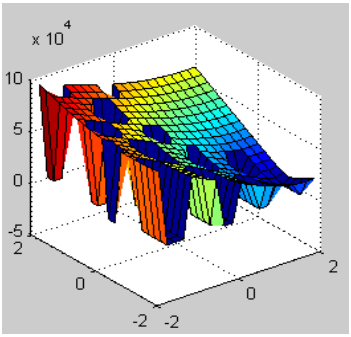


Fig 2 Plot of the Levy No.5 after the first stage

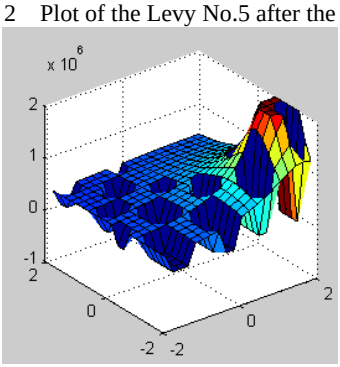


Fig 3 Plot of the Levy No.5 after the second stage

4. REPULSION TECHNIQUE

Stretching technique can be used in the context of QPSO to alleviate the problem of introducing local minima and detect several global minimizes effectively. Difficulties that may arise using these approaches may be overcome through a "repulsion" technique. Input X^*,S,r_{ij},p_{ij} *i=1... N,* and *j=1…m* $X^*=\{X_j^*;j=1,...,m\}$ be the set of already detected minimizes, and $S=\{X_i;i=1,...,N\}$ be the swarm, at a given iteration. The Algorithm is described as follows:

FOR (i = 1: N) Do
If ($X^*\neq 0$) *Then*
$d_{ij}=\|X_i-X_j\|, j=1,...,m$
For (j = 1: m) Do
If ($d_{ij}\le r_{ij}$) *Then*

$$Z_{ij}=\frac{x_i-x_j^*}{d_{ij}}, j=1,...,m$$

$$X_i=X_i+p_{ij}Z_{ij}$$

End If
End For
End If
End For

5. NASH EQUILIBRIUM

A finite k-person game in normal form: There is a set *K={1…k}* of players, each of whom has a strategy set $S_i=\{S_{i1},...,S_{im_i}\}$, consisting of m_i pure strategies [6,7,8]. Each player has a payoff function, $u_i:S\to R$, where $S=S_1\times...\times S_k$.

Let P_i be the set of real valued functions on S_i .For

elements $p_i \in P_i$ we use the notation $p_{ij} = p_i(s_{ij})$ .Let $p \in P, P = P_1 \times ... \times P_k$ and P_i is the set of real valued function on S_i, The set P is isomorphic to R^m .Thus ,we can write $p = (p_1, ..., p_k), p_i = (p_{i1}, ..., p_{im_i}) \in P_i$ .If $p \in P, p_i^{'} \in P_i$,then the notation $(p_i^{'}, p_{-i})$ is used to denote the element $q \in P$, satisfying $q_i = p_i^{'}, q_j = p_j$, for $j \neq i$.

The payoff function u is extended to have domain R^m by the rule $u_i(p) = \sum_{s \in S} p(s) u_i(s)$ where we define $p(s) = p_1(S_1) \times ... \times p_k(S_k)$,A point $p^* \in P$ is a Nash equilibrium of the game, if $p^* \in \Delta$,where $\Delta = \Delta_1 \times ... \times \Delta k$, with $\Delta_i = \{p_i \in P_i : \sum_j p_{ij} = 1, p_i \geq 0\}$,and for all $i \in K$ and all $p_i \in \Delta_i$, the relation $u_i(p_i, p_{-i}^*) \leq u_i(p^*)$ holds.

The computation of Nash equilibrium can be transformed in the problem of detecting global minimizes of an objective function. Specifically ,for any $p \in P, i \in K, s_{ij} \in S_i$,we can define the following functions: $x_{ij}(p) = u_i(s_{ij}, p_{-i})$, $z_{ij}(p) = x_{ij}(p) - u_i(p)$, $g_{ij}(p) = max\{z_{ij}(p), 0\}$

Then ,a Nash equilibrium is a global minimizes of the objective function $v : \Delta \to R$,which is defined by [6]: $v(p) = \sum_{i \in K} \sum_{1 \leq j \leq m_i} g_{ij}^2(p)$

6. EXPERIMENTS AND RESULTS

The "Battle Of Sexes" is a game of two players with two strategies per player. The payoff matrix for each player is $U_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, U_2 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$

The results for the "Battle of Sexes" are reported in Table 1. The proposed approach was also applied on a different game of three players with two pure strategies each, In this case, the payoff matrices are three-dimensional and they are defined as

$U_1^1 = \begin{pmatrix} 0 & -2 \\ -5 & -5 \end{pmatrix}, U_1^2 = \begin{pmatrix} -5 & -5 \\ 1 & -1 \end{pmatrix}, U_2^1 = \begin{pmatrix} 0 & -2 \\ -5 & -5 \end{pmatrix},$

$U_2^2 = \begin{pmatrix} -5 & -5 \\ 1 & -1 \end{pmatrix}, U_3^1 = \begin{pmatrix} 10 & 0 \\ 0 & 5 \end{pmatrix}, U_3^2 = \begin{pmatrix} 0 & 0 \\ 0 & 5 \end{pmatrix}$

The results reported in Table 2.

7. CONCLUSION

In this paper, a quantum version of particle swarm optimization algorithm was introduced. The approach incorporates the recently proposed stretching techniques to overcome local minimizes, as well as a repulsion technique to repel particles away from previously detected minimizes. Experimental results indicate that this is an effective approach for computing Nash equilibrium.

Table 1. Results for computing Nash equilibrium of the game with two players

		Run: 50 iterative: 200		Run: 100 iterative: 500	
		Mean Value	Standard Deviation	Mean Value	Standard Deviation
r-0.5 p=0.8	QPSO	(0.6596, 0.7221)	(1.7536, 1.5447)	(0.6675, 0.6604)	(1.6863, 1.9366)
	QPSO+stretching	(0.7362, 0.7441)	(1.4910, 1.4504)	(0.6551, 0.6654)	(1.6850, 1.6349)
r=3 p=5	QPSO	(0.67771, 0.6212)	(1.7254, 1.5584)	(0.6283, 0.6623)	(2.4801, 2.5518)
	QPSO+stretching	(0.6306, 0.6655)	(1.6283, 1.5287)	(0.6447, 0.6590)	(2.1912, 2.3419)

Table 2. Results for computing Nash equilibrium of the game with three players

		Run: 50 iterative: 200		Run: 100 iterative: 500	
		Mean Value	Standard Deviation	Mean Value	Standard Deviation
r-0.5 p=0.8	QPSO	(-3.3539, -3.3539, 0.2783)	(5.5244, 5.5244, 6.2311)	(-3.3545, -3.3545, 0.2318)	(9.2348, 9.2348, 7.5573)
	QPSO+stretching	(-3.3486, -3.3486, 0.5789)	(4.3980, 4.3980, 4.4920)	(-3.3693, -3.3693, 0.3203)	(7.5383, 7.5383, 7.1452)
r=3 p=5	QPSO	(-3.3539, -3.3539, 0.2783)	(5.5244, 5.5244, 6.2311)	(-3.3545, -3.3545, 0.2318)	(9.2348, 9.2348, 7.5573)

	QPSO+stretching	(-3.1400, -3.1400, 0.4451)	(5.1656, 5.1656, 6.0861)	(-3.2488, -3.2488, 0.3835)	(8.4279, 8.4279, 9.3699)

8. REFERENCES

[1] J. Sun, W.B. Xu, "A Global Search Strategy of Quantum-behaved Particle Swarm Optimization", Proceedings of IEEE conference on Cybernetics and Intelligent Systems. 2004, pp.111 – 116.

[2] J. Sun, B. FENG, et al, "Particle Swarm Optimization with Particles Having Quantum Behavior", [J] Proceedings of 2004 Congress on Evolutionary Computation. 2004, pp.325-331.

[3] J. KENNEDY, R. EBERHART, "Particle swarm optimization", Proceedings of the IEEE International Conference on Neural Networks, 1995, pp.1942-1948.

[4] K.E. Parsopoulos, V.P. Plaglanakos, et al, "Improving the Particle Swarm Optimizer By Function 'Stretching' ". Proceedings of Advances in Convex Analysis and Global Optimization, pp. 29-40.

[5] E.P. Konstantinos, N.V. Michanel, "On the Computation of All Global Minimizes Through Particle Swarm Optimization", IEEE Transactions On Evolutionary Computation. June 2004, 18(3): 211-223.

[6] R.D. McKelvey, A. McLennan, "Computation of equilibrium in finite games," in Handbook of Computation Economics, ser, Handbooks in Economics (13), H.M. Amman, D.A.Kendrick, J. Rust, Eds. Amsterdam, The Netherlands: North-Holland, 1996, vol.1, pp.87-142.

[7] R.D. Luce, H. Raffia, Games and Decisions. New York: Dover, 1985.

[8] M.J. Osborne, A. Rubinstein, A Course in Game Theory. Cambridge, MA: MIT Press, 1994.

A Generic Strategy for Dynamic Load Balancing of Dynamic Parallel Distributed Mesh Generation

Youwei Yuan1,2, Qingqing Guo1
1. School of Computer Science & Technology, Wuhan University of Technology, Wuhan, 430063, China
2. Department of Computer science & technology, Hunan University of Technology, ZhuZhou, Hunan, 412008,China
Email: y.yw@163.com

ABSTRACT:

Parallel mesh generation is a relatively new research area between the boundaries of two scientific computing disciplines: computational geometry and parallel computing. In this paper, we present a new approach for parallel generation and partitioning of 3-dimensional unstructured meshes. The new approach couples the mesh generation and partitioning problems into a single optimization problem. Traditionally these two problems are solved separately with I/O and data movement overheads that exceed 90% of the total execution time for generating, partitioning, and placing very large meshes on distributed memory parallel computers.

Keywords: parallel computing; mesh database; adaptive unstructured meshes; load balancing.

1. INTRODUCTION

Continuous physical systems, such as the airflow around an aircraft, the stress concentration in a dam, the electric field in an integrated circuit, or the concentration of reactants in a chemical reactor, are generally modelled using partial differential equations. To perform simulations of these systems on a computer, these continuum equations need to be discretised, resulting in a finite number of points in space (and time) at which variables such as velocity, density, electric field are calculated. The usual methods of discretisation, finite differences, finite volumes and finite elements, use neighbouring points to calculate derivatives, and so there is the concept of a mesh or grid on which the computation is performed [1,2].

There are two mesh types, characterised by the connectivity of the points. Structured meshes have a regular connectivity, which means that each point has the same number of neighbours (for some grids a small number of points will have a different number of neighbours). Unstructured meshes have irregular connectivity: each point can have a different number of neighbours.

The reasons for parallel mesh generation are twofold. The first one is to reduce the computational times required for the generation of large grids. The second one, and more important, is to avoid the very large single-processor memory requirements in sequential generators.

In this paper, we propose a parallel progressive Delaunay mesh generation algorithm, which achieves a good load balance with minimal communication overhead [3,4,5]. The method is fully distributed and is based on a very simple, low-overhead load-balancing algorithm that runs in every node. We prefer a distributed method because it would have less chance of running into bottleneck problems and is generally more reliable. The method requires no global information, and no broadcasting information. Through actual implementations, the net gain in performance due to the use of this method is found to be substantial and comparable to performance results in the literature for decentralized remapping [3]. Our approach is to identify the hierarchical computational structure in this problem to exploit data parallelism instead of control parallelism. The processor assignment for irregular patch objects uses a novel object ordering with cyclic mapping to achieve load balance even if computation associated with patch objects changes [6,7,8].

This paper is organized as follows. Section 2 gives preliminary definitions. Section 3 presents parallel mesh refinement program. Section 4 presents the performance and conclusions in section 5.

2. PRELIMINARY DEFINITIONS

2.1 The Definitions of Mesh Generation

Let a set of the input scattered points $p_i(x_{1i},x_{2i},x_{3i})(i=1,\ldots,n)$ be given. Coordinate of the output points: $q_j(\omega_{1j},\omega_{2j},\omega_{3j})$ *(j=1,…,m)* .Each cell C_j in the network has a weight vector whose dimensionality is the same as that of the input vectors.

As shown in Fig. 1., the desired element size at a vertex *P* of a given surface mesh *T* is represented by a real value $\left\|\overrightarrow{PQ}\right\| = h(p)$ characterizing a posteriori the ideal length (*i.e.* the unit length) of any edge *PQ* sharing *P*. Let M_i is the (discrete) size map associated with the vertices of *T*. By interpolating M_i over *T*, we obtain a continuous size map in R^i, denoted as M_i^c .

Definition 2.1 *Mesh T conforms to the map* M_i *iif*

$$\forall PQ \in T, \quad \frac{1}{\sqrt{2}} \le l_{PQ} \le \sqrt{2} \qquad (1)$$

Where l_{pq} *is the length of edge PQ with respect to* M_i^c .

If the edge *PQ* is parameterized between 0 and 1 by $P + t\overrightarrow{PQ}$, its normalized length is then expressed as

$$l_{PQ} = \left\|\overrightarrow{PQ}\right\| \int_{\delta}^{i} \frac{1}{H_{PQ}(t)} dt \qquad (2)$$

Where $H_{PQ}(t)$ is a monotonous element size interpolation function such that $H_{PQ}(0)=h(p)$ and $H_{PQ}(1)=h(Q)$. *h(p)* and *h(Q)* being the desired element sizes at *P* and *Q*.

A geometric size map g_i can be defined, such that the gap value between the triangles of a mesh and the surface is

controlled. At any mesh vertex *P*, the minimum of the principal radii of curvature r_p can be approached using the following trivial formula (cf.Fig. 1.)

$$r_P = \frac{1}{2}\min_{P_i}\frac{(\overrightarrow{PP_i},\overrightarrow{PP_i})}{(v_P,\overrightarrow{PP_i})} \quad (3)$$

Where P_i covers the set of the endpoints of all edges sharing *P* and v_p is the normal to the surface at *P* and <., .> denotes the dot product.

Definition 2.2 A *mesh satisfying the rectified geometric size map is a geometric mesh in which the elements are all well shaped.*

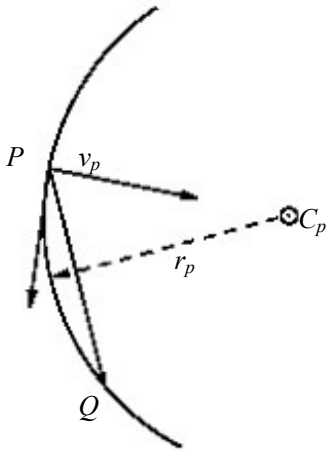
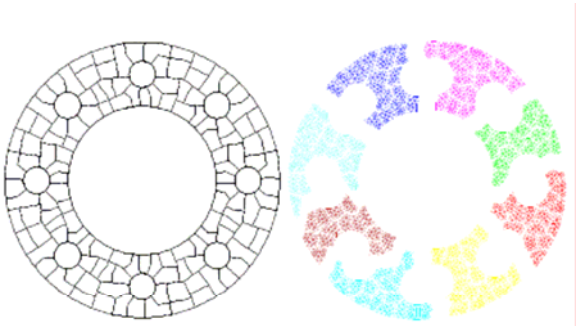


Fig. 1. Approximation of the minimum of the principal radii of curvature rp at P



(a) continuous geometry (b) domain

Fig. 2. Original geometry and domain decomposition for parallel Computing

2.2 Domain Decomposition for Parallel Mesh Computing

There are two distinct phases to creating a mesh, these being global/sequential and local/parallel respectively [9,10]. The global part of the mesh creation involves resolving the topology of the problem domain by splitting it into a small number of domains, each of simple topology and geometry, such as hexahedra or tetrahedra [11,12]. This procedure requires global knowledge of the geometry, is logic intensive, and deals with a small amount of data, and is thus suitable for a sequential machine [13,14,15].

The key step is the partitioning step, in which a set of edges in the final mesh is determined by a projection method prior to generating any elements. The input is then split and redistributed among the processors in each team. Fig. 2. shows the original geometry and domain decomposition for parallel Computing, decomposition for parallel computing.

3. PARALLEL MESH REFINEMENT PROGRAM

3.1 The Theoretical Framework

Sequential Delaunay refinement algorithms are based on inserting circumcenters of triangles which violate the required bounds,e.g. the upper bound $\bar{\rho}$ on circumradius-to-shortest edge ratio, and the upper bound $\bar{\Delta}$ on triangle area. Let the cavity CM (p) of point p with respect to mesh M be the set of triangles in M, whose open circumdisks include p. We expect our parallel Delaunay refinement algorithm to insert multiple circumcenters concurrently in such a way that at every iteration the mesh will be both conformal and Delaunay.

Theorem 1 Let $\bar{r}$ be the upper bound on triangle circumradius in the mesh and pi; pj$\in \Omega \subset$ R2. Then if // pi -pj // $\geqslant 4\bar{r}$, then independent insertion of pi and pj will result in a mesh which is both conformal and Delaunay.

3 .2. Parallel Mesh Refinement Program

We show that a simple 2D algorithm with this modification generates a constrained Delaunay triangulation (CDT) of the input domain in which every triangle t has a circumradius to shortest edge ratio (t) no greater than p2. Further, as a result of allowing poorly shaped triangles to be refined in any order.

Theorem 1 Let $d = min_{p\in Q}\ lfs(p)$. The following invariants hold after each iteration though Main Loop:

1. *Each refined subsegment has length at least 2 .d*
2. *Each refined subfacet has circumradius greater than* $d\sqrt{2}$

Algorithm .1 shows parallel mesh generation/refinement program. Where *np* denotes number of input points and *ne* denotes number of edges.

```
Initializations
      num=0
      newnum(1:ne)=0
For i=1,np
      For j=1,ne
            IF P_i is a vertex in k_j and newnum(j)=0
                  num=num+1
                  newnum(j)=num
for each processor P_i do
  for I=1,N/P do
     S_i ← read ith subdomain
     Create amobile object S_i^O
endfor
for j=1,K do
  Register_Task T_j
for I=1,N/P do  /* Perform Parallel Computation: */
  for j=1,K do
Remote_Service_Request(T_j--> S_i^O)
```

4. PERFORMANCE

The example includes parallel Delaunay mesh generation and mesh refinement. We also perform dynamic load balancing on a fixed mesh with variable processing resources .The parallel version uses the MPI communications library although we are working on a shmem version [1] which could be expected to show even faster timings.

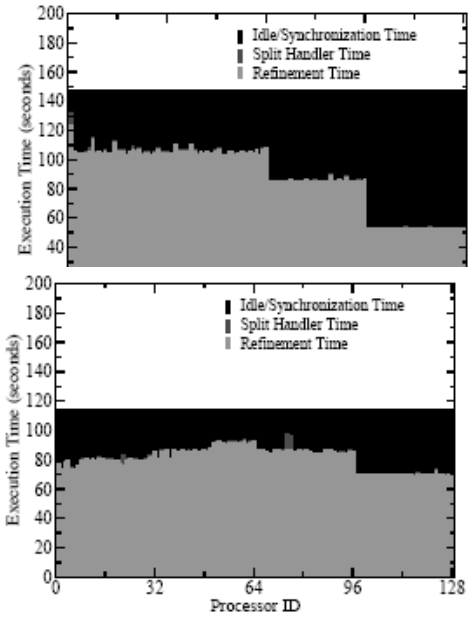


Fig. 3. The execution time for the cross section of the rocket pipe whose data are equidistributed on 128 heterogenous processors; without load balancing (a) and with load balancing using parallel mesh generation (b)

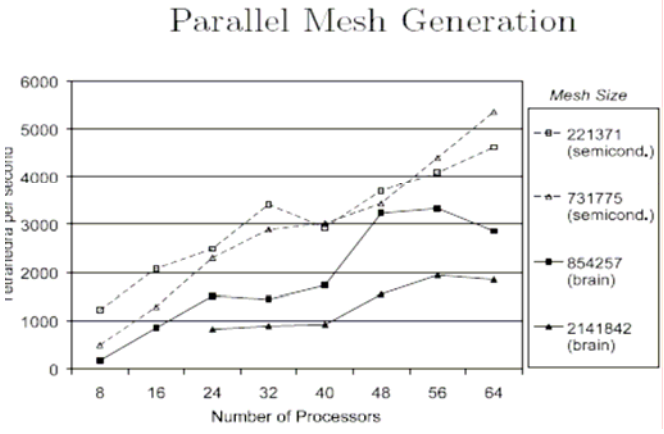


Fig. 4. Surface of the tetrahedral mesh for a simplified model of a human brain generated from an advancing front method

Fig. 3. shows the impact of dynamic load balancing on the performance for the cross section of dinosaur. Although we used state-of-the-art ab-initio data partition methods for equidistributing the data and thus the computation among all 128 processors, the imbalances are due to heterogeneity of the three different clusters; the first 64 processors are from Typhoon (slowest cluster), the next 32 processors are from Tornado and the last 32 processors are from Whirlwind (the fastest cluster). Again, the dynamic load balancing improved the performance of parallel mesh generation by 23%. Finally, the data from Fig. 3. Indicate the impact of work load imbalances due to: (1) the differences in the work-load of submeshes and (2) heterogeneity of processors using the PTE method. Fig. 3. (b), shows that the speed of the PTE method is substantially lower, for the brain model (see Fig. 3. a), due to work-load imbalances; while for a more regular geometry (the semiconductor test case [5]).

Fig. 4. shows surface of tetrahedral mesh for a simplified model of a human brain generated from an advancing front method.

5. CONCLUSION

In this paper, we propose a parallel progressive Delaunay mesh generation algorithm, which is fully distributed and is based on a very simple, low-overhead load-balancing algorithm that runs in every node. We have described a new method for optimizing and load balancing graph partitions with a specific focus on its application to the dynamic mapping of unstructured meshes onto parallel computers.

6. ACKNOWLEDGEMENT

The work was supported by NSFC (Grant No: 60173046) and also supported by the nature science foundation of Hubei province (No: 2005ABA227) and the nature science foundation of Hunan province (No:05JJ40111).

7. REFERENCES

[1] A. Arulananthan, S. Johnson, et al, A generic strategy for dynamic load balancing of distributed memory parallel computational mechanics using unstructured meshes. Proc. Parallel CFD ', 1997，102-108.

[2]Y.W. Yuan, L.M. Yan, et al, "Efficient surface mesh reconstruction from unorganized points using neural networks", Chinese Journal of electronics, 2005, 14（1）: 26-29.

[3] C. Z. Xu, F. C. M. Lau, "The generalized dimension exchange method for load balancing in k-ary n-cubes and variants", J. Parallel Distrib. Comput. 1995，24（1）:72–85.

[4] G. C. Fox, "Applications of parallel supercomputers: Scientific results and computer science lessons", in Natural and Artifical Parallel Computation, The MIT Press, Cambridge,MA, 1990.

[5] M. J. Berger, S. H. Bohari, A partitioning strategy for nonuniform problems on multiprocessors,IEEE Trans. Comput., 1987，36（5）:570–580.

[6] P. J. Frey, Generation and adaptation of computational surface meshes from discrete anatomical data. International journal for numerical methods in engineering. 2004, 60（2）: 1049~1074.

[7] L. P Chew， R. L. Drysdale, Based on convex distance functions.TechnicalReport.Tech.Report PCSTR 86-132，Dept.of Computer Science，Darmouth College，November, 1986.

[8] G. M. Nielson，T. A. Foley, A survey of applications of anaffineinvariannorm.In:Lyche T， Schumaker LL， editors，Mathematics methods in CAGD，Boston，MA:Academic Press，1989， 445-467.

[9] OkabeA，BootsB et al，Concepts and applications of

Voronoi diagrams.New YOrk:Wiley， 1992.
[10] D，Azzevedo E F et al，Optimal interpolation triangle incidences. SIAM Joumal of Science and Statistical Computation, 1989，10(6):1063-1076.
[11] D. Mavriplis, Adaptative mesh generation for viscous flows using Delaunay triangulation. Journal of Computational physics.1990 (90): 271-291.
[12] M. K. Posenau, Approaches to high aspect ratio triangulations、Proceeds of sth canadian Conference on Comp.Geom.Waterloo，ontario，1993.
[13] Per-olofFjallstrom, Evaluation of a Delaunay-based method for surface approximation， Computer-Aided Design. 1993，25（11）:711-719.
[14] A.F. Michael, Inplementation of arandomized algorithm for Delaunay triangulations in three dimensions， Computer-Aided Geography Design.1995 (12), 349-370.
[15] IslerV，R.W.H. Lau et al, Real-time multi-resolution Inodeling for complex virtual environments.In proc Of VRSI’96. HongKong，1996.

Solving linear systems on global computing architecture with GMRES method

Haiwu HE1,2, Guy BERGERE2，Zhijian WANG1, Serge PETITON2
1: Computer and Information Engineering School,
HOHAI University,Nanjing,JiangSu,China
2 : Laboratoire d'Informatique Fondamentale de Lille
INRIA Grand-Large Project, USTL
Villeneuve d'Ascq, France
Email : hehaiwu@hhu.edu.cn, bergereg@lifl.fr

ABSTRACT

By making use of a very large amount of unexploited computing resources, grid computing achieves high throughput computing. We present a classical parallel method GMRES (m) to solve large sparse linear systems utilizing a lightweight GRID system XtremWeb. XtremWeb is a global computing platform, which is dedicated to multi-parameters generic applications. We have implemented this important algorithm GMRES (m) which is one of the key methods to resolve large, non-symmetric, linear problems on this system. We discuss as well the performances of this implementation deployed on two XtremWeb networks: a local network with 128 non-dedicated PCs in Polytech-Lille of University of Sciences and Technologies of Lille in France, a remote network with 3 clusters of SCGN Grid including 91 CPUs totally in the High Performance Computing Center of University of Tsukuba in Japan. We also do the tests on the platform of supercomputer IBM SP4 of CINES in Montpellier in France. We compare the performances on the two different computing systems. The advantages and drawbacks of our implementations on this GRID computing system XtremWeb will be well explained.

Keywords: global computing, XtremWeb, GMRES, supercomputing, lightweight grid system.

1. INTRODUCTION

Nowadays, many devices with wires or wirelessly interconnected by huge Internet such as the clusters, the PCs, even the PDAs, handsets, mobile phones could be widely scattered geographically. The main profit of Global Computing is to use their idle time to run a very large and distributed application. And the volunteer devices and instruments, which bestow voluntarily some time of their unexploited time to execute a part of application with the associated data, provide the computing power. In this mode, the computing resources are in fact a heterogeneous gathering of GRID. In many technical aspects every devices or instruments could be different greatly: CPU speed, CPU load, storage space, hardware configuration, operating system, and network bandwidths, network load etc. Some implementations have already done, which include the Globus, SETI@HOME project, XtremWeb etc[3,4,5,6,8].

XtremWeb project aims at building a platform for studying execution models in the general framework of Global Computing. Just as some Global Computing projects SETI@home [6] or Folding@home, the goal of XtremWeb is to distribute applications over a set of devices in Internet or Intranet via cycle stealing scheme and focus particularly on multi-parameters applications which can be run many times with different input parameters. Every computing component is autonomous from each other's totally [8].

The IBM RS/6000 SP series supercomputer systems are widely used in the world for various scientific and commercial applications. In the list of TOP 500 supercomputers, the IBM RS/6000 SP machines have a leadership presence. Equipped with high-speed processors (Power 4), large bandwidth of interconnections between the nodes (HPS-Federation Switch), combined with its commercial or free scientific software and latest AIX 5.2 Unix system, the supercomputer system IBM SP4 of CINES (Centre Informatique National de l'Enseignement Supérieur) in France provides a distinguished scientific calculation environment.

For the resolution a nonsymmetric linear system of equations $Ax = b$, the GRMES algorithm is a classic iterative method. Saad and Schultz explained in [9] this popular GMRES method. They give a practical implementation GMRES on the base of Arnoldi process and Givens rotations. An enhanced version with restart after *m* steps GMRES (*m*) is as well presented in [9], this algorithm allows computing sparse matrices in compressed formats, without loading zeros which are of no use for the computation in memory. Furthermore it keeps sparse structure, which avoids supplementary communications [1],[12]. It has been well implemented in parallel environments [10].

In this article, we present a distributed version of GMRES (m) algorithm, which is well implemented on the lightweight GRID system XtremWeb. Moreover we will compare the performances in XtremWeb with those of IBM SP4 in a supercomputing context. We will find the good features of XtremWeb: heterogeneous, good fault tolerance, high throughput, large-scaled, and low cost.

2. GMRES (M) ALGORITHM INTRODUCTION

The **GMRES** (**G**eneralized **M**inimum **RES**idual) method was proposed by Saad and Schultz in 1986 see [9] as a Krylov subspace method for solving non-symmetric systems. The $m^{th}(m \geq 1)$ iterate x_m of GMRES is the solution of the least squares problem

$$\min_{x \in x_0 + k_m(A, r_0)} \left\| b - Ax \right\|_2 \qquad (2.1)$$

$$r_0 = b - Ax_0$$

where is the residual of the initial solution.

The Arnoldi process applied to $K_m(A,v)$, builds $V_{m+1} = [V_m, V_{m+1}]$, and $\overline{H}_m$.

Since $K_m(A,v) = span(v, Av, ..., A^{m-1}v)$, the iterate x_m has the following form

$$x_m = x_0 + V_m y_m$$

where $y_m \in \Re^m$, and its residual r_m can be written as following

$$\begin{aligned} b - Ax_m &= b - A(x_0 + V_m y_m) \\ &= r_0 - AV_m y_m \\ &= V_{m+1}(\beta e_1 - \overline{H}_m y_m) \end{aligned}$$

where e_1 is the first canonical vector $\Re^{m+1}$.

Now we can compute the norm of r_m, and we have

$$\begin{aligned} \|r_m\|_2 &= \|V_{m+1}(\beta e_1 - \overline{H}_m y_m)\|_2 \\ &= \|\beta e_1 - \overline{H}_m y_m\|_2 \end{aligned}$$

Therefore, y_m is the solution of the following least squares problem:

$$\min_{y \in \Re^m} \|\beta e_1 - \overline{H}_m y\|_2 \qquad (2.2)$$

A powerful tool for solving this optimization problem is the QR decomposition based on Householder transformations [14].

In the GMRES algorithm the number of vectors requiring storage increases with m. One way to address this problem is using the algorithm iteratively, by finding the iterate x_m, and restarting the algorithm with the initial guess $x_0 = x_m$, until convergence. Thus, we obtain the restarted GMRES (m) after m iteration of GMRES.

Algorithm: GMRES (m)

1. Start: choose x_0 an initial guess of the solution,

m The dimension of Krylov subspaces, and ε the tolerance,

Compute $r_0 = b - Ax_0$.

2. Apply Arnoldi process to $K_m(A, r_0)$.

3. Compute $y_m = \arg\min_{y \in \Re^m} \|\beta e_1 - \overline{H}_m y\|_2$ with QR factorization, and set $x_m = x_0 + V_m y_m$, $r_m = b - Ax_m$.

4. Restart: if $\|r_m\|_2 \le \varepsilon$ stop

Else set $x_0 = x_m, r = r_m$, and goto 2

And set $x_m = x_0 + V_m y_m$, $r_m = b - Ax_m$.

3. XTREMWEB GRID SYSTEM

The lightweight desktop GRID system XtremWeb [3],[5],[8],[11] which intends to distribute applications to dynamic and volatile resources matching up to their availability by utilizing its own fault tolerance and security policies. It is an Open Source, Free Software (GPL) and non-profit software platform for scientific applications of Global Computing and Peer to Peer distributed systems [11].

In XtremWeb framework, to join in the global computing, the devices or instruments can play two roles:

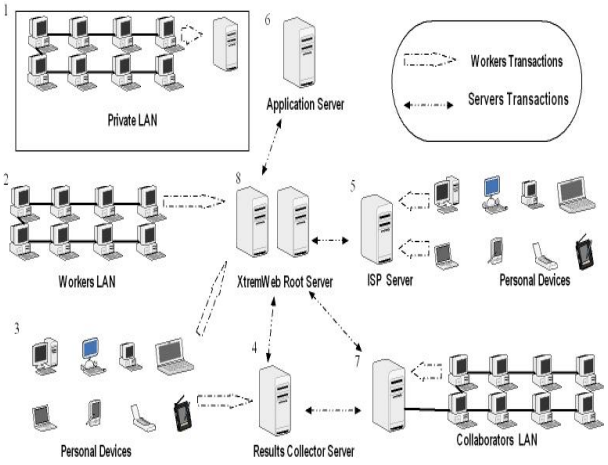


Fig. 1. XtremWeb Framework outline

A Collaborator: A Collaborator is powerful machine that registers to XtremWeb administration server. It can download the whole XtremWeb system software, and then setup its own global distributed computing environment. The Collaborator gives an accord to XtremWeb administration server. This accord allows XtremWeb main server to exploit the surplus resources, which are collected by the Collaborator. In this pattern the group of devices governed by a Collaborator will do computing jobs distributed by main XtremWeb server only when they are idle and no jobs allocated by their Collaborator.

A Volunteer: A Volunteer is a device or instrument that registers to the XtremWeb administration server voluntarily. It downloads the computing components and installs them. While it has idle time, it will contribute it for the computing jobs.

An outline of XtremWeb global computing framework is given in 0. In the centre of this framework is XtremWeb Root Server (8). It distributes the computing jobs and administrates the communications. A local network of tens even hundreds of PCs in a school or an institution connected directly with XtremWeb server donate its idle time collected to global computing (2). Collaborators can manage their own distributed applications (6, 7) and work together with XtremWeb

while they have spare resources to share. Some server can be reconfigured for special purpose such as collecting results (4). A more universal pattern (5) is to do global computing via an ISP (Internet Service Provider) server, which connects with millions of personal devices (PC, Laptop, PDA, iMac, Palmtop, Digital handset etc). It is a ideal commercial pattern for GRID computing, because the computing time consumed for each device can be recorded then paid by ISP. We can as well connect directly without any intermediate server these various personal devices with XtremWeb server to do our computing (3). Anyhow, a school, an enterprise or an institution can setup its completely isolated Intranet XtremWeb system for their private computing jobs (1).

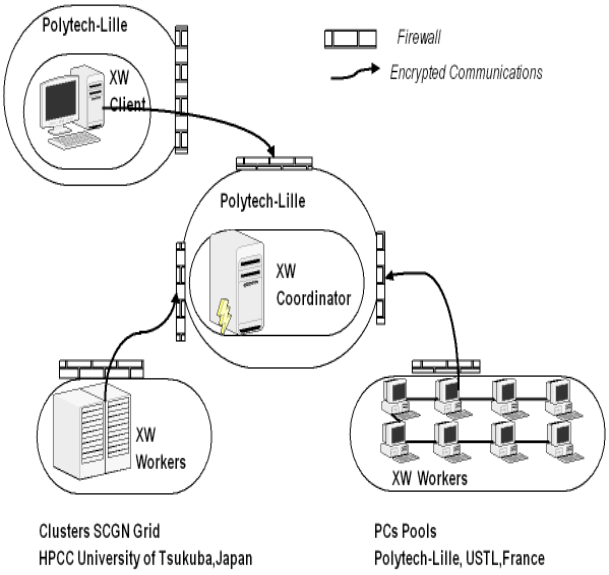


Fig.2. XtremWeb Network Implementation Overview

The tasks management mechanism of XtremWeb system is the Coordinator-Worker mode (see 0). The Coordinator administrates the tasks management process. Especially, it reserves all the results, but unfortunately no true concept of tasks scheduling is applied until now. There is only the FIFO (First In First Out) scheduling policy. Workers are the dispersed volunteer devices, which would like to donate their unexploited CPU time to execute the computing jobs provided by the Coordinator. It is a "pull" model: Workers only initiate all actions and connections. In the beginning, every Worker connection is registered by the Coordinator. According to its local policy, Worker demands tasks from the Coordinator to execute the applications. For network security all of the communications between Coordinator and Workers are well encrypted

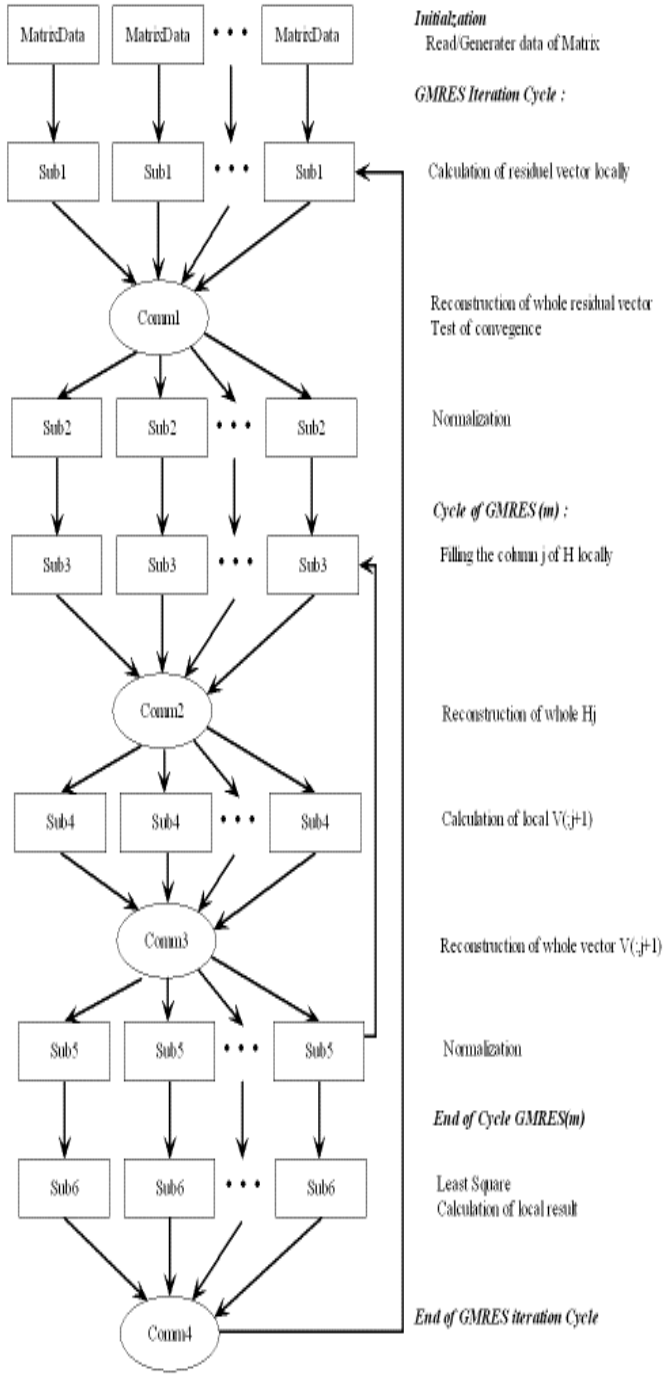


Fig 2. GMRES (m) Implementation Schema on XtremWeb.

Worker downloads the executable components and all the associated data (the input files, the command line arguments for the executable binary file, etc), saves them on local storage media then begins the computing. Once a task terminated, Worker sends the results (the output files) to the Coordinator.
For network security, to guarantee user authentication, Workers integrity, application and results protection and user execution

logging, XtremWeb relies on three mechanisms: the authorized users list, the Coordinator authentications by Workers and Clients, the sandbox utility. There are at the same time the firewalls on every site to assure the network security. XtremWeb protocols use single side (the pull model of Worker side) communications to go through the firewall.

4. XTREMWEB IMPLEMENTATION OF GMRES (M)

In our implementation of GMRES (*m*) on XtremWeb system, for saving the memory and cutting communications on the network that all the sparse matrices are stocked in the compressed format **CSR**. The most economical format for the sparse matrices is the format **CSR** (Compress Sparse Row) or its equivalence by column **CSC** (Compress Sparse Column). This format **CSR** stocks the sparse matrices by three vectors: A (the vector contains all the nonzero elements), JA (the vector contains the number of columns of the element in A), IA (the vector contains the number of nonzero element where begin each line).

To implement this GMRES (m) algorithm, 11 software components are written. In Fig 2 the components with the name beginning with "comm" are executed on Coordinator of XtremWeb system, they are sequential programs. The components with the name starting with "sub" are executed on Workers of XtremWeb in a parallel way. In reality, at a given moment a job will be executed on which Worker is totally decided by the XtremWeb scheduling strategy and Worker local policy.

5. NUMERICAL RESULTS AND ANALYSIS

The first matrix tested is the matrix "vp" with the size $2q \times 2q$ presented in the 0, its eigenvalues are $a_j + ib_j$, and we take the eigenvalues in two rectangles R_1 with vertex $-2.5 \pm 2i$, $-1.5 \pm 2i$, and R_2 with vertex $1.2 \pm 4i$, $1.8 \pm 4i$ [12]. In our tests, we use a matrix "vp" generated with the size 100,000 saved in the Matrix Market formatted file, later we use it as "matrixvp1E05".

$$A = \begin{Bmatrix} a_1 & \frac{b_1}{2} & & & & & & & & \\ 2b_1 & a_1 & & & & & & & & \\ & & a_2 & \frac{b_2}{2} & & & & & & \\ & & 2b_2 & a_2 & & & & & & \\ & & & & \bullet & \bullet & & & & \\ & & & & \bullet & \bullet & & & & \\ & & & & & & a_q & \frac{b_q}{2} \\ & & & & & & 2b_q & a_q \end{Bmatrix}$$

Fig. 4. Matrix Sparse VP

The second matrix is matrix "DA" which is a dense, square matrix generated by a matrix generator. "DA20000" is a matrix with dimension 20,000 including 4×10^8 nonzero elements.

Two XtremWeb network configurations are employed: a LAN and a WAN based configuration. The Coordinator of the two XtremWeb network runs on a dedicated server in the building of engineer school of Lille (Polytech-Lille) in Lille. The local configuration, Workers (128 PCs, non-dedicated, Linux system) run in the computer rooms of Polytech-Lille. The WAN configuration makes use of the 3 clusters of SCGN Grid [17] in High Performance Computing Center of University of Tsukuba in Japan. Totally, 82 Workers (CPUs in the clusters) are in use. The description of details of the two sites is given in the Table 1 . And the powerful configuration of IBM SP4 is shown in Table 2.

The goal of our tests is not to measure the performances on each peer nor of our lightweight grid XtremWeb because we will just obtain a literally worthless set of data. In fact, because of the dynamic and volatile nature of grid, this kind of information is inexplicable and is valuable only for a system with an efficient scheduling in real time. Even so, it is still helpful to explain the final test results for proposing XtremWeb – a good lightweight desktop grid system for some numerical computing resolutions.

Table 1. XtremWeb Network Configuration

Local configuration in Lille (128 PCs)		
Number	CPU	Memory
28	Pentium III (Katmai), 450MHz	128MB
28	Intel Celeron, 2.2GHz	512MB
23	Intel Celeron, 2.4GHz	512MB
15	AMD Duron, 750MHz	256MB
14	Celeron (Coppermine), 600MHz	128MB
8	Intel Celeron, 2GHz	512MB
8	Intel Celeron, 1.4GHz	256MB
4	Pentium 4, 2.4GHz	512MB
WAN configuration in Tsukuba (3 Clusters)		
Number of Node	CPUs of node	Memory per node
	Dual P4 Xeon 3065MHz	1GB
10	Dual Athon 1533MHz	882MB
8	Quad P3 450MHz	2GB

Table 2. IBM SP4 Configuration

IBM SP4 (288 CPUs)		
Number of Node	CPUs of node	Memory per node
2	32 Power 4 1.3GHz	64GB
2	32 Power 4 1.7GHz	64GB
5	32 Power 4 1.7GHz	32GB

In reality, the computing time varies greatly (the difference processor frequency, available memory, computing load, network traffic, etc and as well the high

volatility of Workers of XtremWeb). So all the data used in our results should be save rage values of many tests.

Table 3. MatrixVP1E05 WITH 10 WORKERS ON WAN (Tsukuba) (M (GMRES)=30,Iterations=5,3730 jobs)

	With Data File	With Generator
CPU Time	2.71s	2.89s
Computing Time (Coordinator)	33223s	26252s
Communication Time	61947s	55482s
Total Time	95170s	81734s

The way of storage of our matrices used for our computing is a key factor to the performance for our tests. We use two manners to do it. One is to read data file by all the subroutines in each steps of our algorithm; it is the case when we have to deal with the matrices downloaded from Internet site in Matrix Market or the other formats. The other is to generate the matrices locally by a distributed version matrix generator when we have to use the data of matrix to do some calculation. In fact, in Fig 2, only two components "sub1" and "sub3" call for the matrix to do the multiplication.

In Table 3, for the computing time (CPU time), we can effortlessly notice the modest CPU time consumed because of powerful CPUs of the clusters and the small data volume of this matrix (for one Worker, the whole matrix "matrixvp1E05" is cut into 10 pieces with the dimension of 10,000 including 20,000 nonzero elements). We can as well notice that we gain 15% performance in this case of generator configuration thanks to significant diminution of communications and traffics in networks.

Table 4. DA20000 on LAN (N=20000. NNZ=20000*20000) Lille with Generator

	10	15	20	25	30
Jobs	800	1200	1600	2000	2400
Memory(M)	480.7	320.6	240.6	192.6	160.5
CPU Time	45.88s	22.37s	11.21s	4.71s	3.97s
Computing Time (s) (Coordinator)	3371.6	2291.6	1429.0	718.7	1697.7
Communication Time(s)	706.7	1027.7	1423.9	1893.3	2444.6
Total Time(s)	4078.3	3319.3	2852.9	2612.0	4142.3

Results shown in Table 4 reveal that when we increase nW (the number of Workers participating the global computing), we obtain a finer granularity, which means for every job executed, the data to compute are less, then we consume less CPU time of computing, meanwhile the number of jobs augments linearly with more Workers. The CPU time consumed for computing decreases with more Workers thanks to smaller volume of data to compute. For total time that we evaluate on the Coordinator, there is an optimal value nW=25 in Table 4, when nW is 25, we have the shortest computing time on Coordinator side. When nW is small, the principal time is utilized for the computing, in this table "nW=10, 15" are those cases. But the network traffic increases greatly with more Workers involved in. We can remark the total communication time increases much when we use more Workers for our tasks. In all, the time to get convergence for our GMRES (m) method increases with the increase of Workers after the optimal threshold (nW=25). This result matches to the theory that the parallel implementation of GMRES (m) requires intensive communications and multiple synchronizations.

Table 5. DA20000 (N=20000. NNZ=20000*20000) IBM SP4 with Generator

	10	15	20	25	30
CPU Time	12.88s	8.67s	7.26s	6.13s	4.74s
Communication Time	15.26s	12.53s	12.25s	10.85s	7.95s
Total Time	28.14s	21.20s	19.71s	16.98s	12.69s

We also do some tests for the same matrix on the supercomputer platform IBM SP4; the results are shown in Table 5. Compared with the tests done in Table 4, we observe the remarkable difference. When the number of processors or Workers involved in is from 10 to 20, the CPU time consumed for IBM SP4 is less, which can be explained by the limitation of hardware configuration of PCs in the LAN of Polytech-lille. In fact, even in our tests for better performance, we make Workers run on the more powerful PCs (memory $\geq$ 512M, main frequency $\geq$ 2GHz), because of limitation of memory and shared with other applications running on the Workers when memory demanded is larger than 200 M, it causes frequent paging in hard disk. Finally, the latency of CPU becomes longer, even the more powerful processors of LAN than IBM SP4 (in Table 2, main frequency $\leq$ 1.7GHz). But when more Workers (nW=25, 30) involved in, the demand of memory is much less, the CPU time used for computing of LAN is less than that of IBM SP4. It can be explained by extra cost of the function MPI_WTIME () that is used for IBM SP4 MPI to get the universal time and includes the additional time for synchronization.

The communication time in Table 4 keeps increasing with more Workers joined in the global computing. Contrarily, that of IBM-SP4 decreases continuously with more processors engaged in (see Table 5). And the time of communication of IBM SP4 is much less thanks to the very high speed connection (Federation Switch US protocol on same node 1201Mb/s [18] between the processors of this supercomputer, however the maximum bandwidth of LAN in Polytech-lille is only 10Mb/s. And because of excellent performance of network hardware of supercomputer IBM SP4, when we augment the number of processors for parallel computing from 10 processors to 30 processors, the time of communication decreases correspondingly, in all we need less time to convergence for GMRES method.

6. CONCLUSION

We implemented our GMRES (m) algorithm in two computing platform: A desktop grid system XtremWeb, a supercomputer system IBM SP4. We could tell the differences. Without doubt, the supercomputer system IBM SP4 has the much better

performance thanks to its excellent hardware and software configuration. But it has a key shortcoming: too expensive. A high school or a small company cannot afford it.

The lightweight GRID computing system XtremWeb we present principally in this paper is really a good system for GRID computing. Compared with the performance of Supercomputers, the time of resolution is much more. But the interest of the grid-computing model is to use the inexpensive PCs in computer room or at home interconnected by Internet and free time of these unexploited resources. XtremWeb is well designed for profiting it. Facing the volatile character of global computing network with thousands of even more devices interconnected by Internet, XtremWeb has a good competence of fault tolerance at the same time. In fact, additionally in our practice, the processors of IBM SP4 are limited (288 CPUs maximum), and when we demand the utilization of processors of several nodes ($nP \geq 32$), because of many users sharing the expensive resources; our tasks may be always in the queue of waiting for execution for a few days even weeks. But for XtremWeb, we can launch our jobs as we want, and more than thousands of processors can be engaged in, usually our jobs are executed as we launch thanks to persistent availability of idle time of devices. Truly until now the LAN of Polytech-Lille is not fully exploited, there are few users of P2P platform to profit this computing resource. That makes us easier to accomplish our computing tasks than on the Supercomputer IBM SP4.

But we observe equally that the sequential components executed on Coordinator of our GMRES computing software (see Fig 2) along with the central management of all communications turn out to be the bottleneck, the communication can be optimized and implemented in a decentralized mode. And a more efficient scheduling instead of FIFO mode for the jobs distributed to Workers would improve significantly the performance. Even so, we anticipate that longer lasting computing jobs will obtain better performance with relatively low charge of communication. And for the parallel algorithm like GMRES (*m*), which necessitates a lot of communications and synchronizations, the XtremWeb is not a very ideal computing platform.

Our results of tests confirm moreover the portability of our implementation on XtremWeb system. In future, with more Workers and with larger bandwidth, we will resolve the very large-scale linear systems with the grade of millions without difficulty.

7. REFERENCE

[1] H.W.He, Guy Bergère, Serge PETITON, "A parallel asynchronous hybrid method to accelerate convergence of a linear system", Proceeding of International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2004.

[2] L. Aouad, S. Petiton, "Experimentations and Programming Paradigms for Matrix Computing on Peer to Peer Grid", Proceeding of the 5th IEEE/ACM International Workshop on Grid Computing (GRID'04), 1550-5510/04 IEEE.

[3] G. Fedak, C.Germain, et al "XtremWeb : A Generic Global Computing System", 0-7695-1010-8/10, 2001 IEEE.

[4] I. Foster, C. Kesselman, "The Globus Toolkit", the grid: blueprint for a new computing infrastructure, 1998, 259-278.

[5] F. Cappello, S. Djilai, et al, "Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with grid", Future Generation Computer System, 2005,21:417-437.

[6] David P. Anderson, Jeff Cobb, et al, "SETI@home: an experiment in public-resource computing", Communications of the ACM, 2002,45(11): 56-61,2002.

[7] Internet software consortiom and network wizards – Internet domain survey, July 2004.

[8] O.Lodygensky, G. Fedak, et al, "XtremWeb & Condor: sharing resources between Internet connected Condor pools", GP2PC2003, Global and Peer-to-Peer Computing on Large Scale Distributed Systems, Tokyo, Japan, 2003.

[9] Y. Saad, M. H. Schultz, "GMRES: A Generalized GMRES Algorithm for Solving Nonsymmetric Linear Systems", SIAM J. Sci. Statist. Compt., 1986,7:856-869.

[10] R.D.Da Cunha, T.Hopkins, "A parallel implementation of the restarted GMRES iterative algorithm for nonsymmetric systems of linear equations", advances in Computational Mathematics, 1994,2:261-277.

[11] XtremWeb User Guide, Installation, Usage and Programming, Edition 1.3.0, for XtremWeb version 1.3.0.

[12] Guy BERGERE, Contribution à une programmation parallèle hétérogène de méthodes numériques hybrides, thesis 1999.9. USTL

[13] G.T.KELLEY, Iterative methods for linear and nonlinear equations, no.16 SIAM Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[14] G.H.GOLUB, C.F.VAN LOAN, Matrix Computations, The Johns University Press, Baltimore, third edition, 1996.

[15] R.S.Martin, G.Peters, et al. "The QR algorithm for real Hessenberge matrices", Numerical Mathematics, 1970.

[16] S.Petiton, "Parallel QR algorithm for iterative subspace methods on the connection machine (CM2)", In J.Dongarra, P.Messina, D.Sorensen, and R.Voigt, editors, Parallel Processing for Scientific Computing, SIAM, 1990.

[17] http://www.omni.hpcc.jp/ganglia/

[18] IBM SP Systems Overview, IBM Corporation, http://www.llnl.gov/computing/tutorials/ibm_sp/#Switch

A kind of the Cluster Computing Scheduling Algorithm Based on LAN *

Qing Yang, Yan Hu, Ge Wang, Shijue Zheng
Department of Computer Science, Central China Normal University, Wuhan,
Wuhan, Hubei 430079, P.R.China
yangq@mail.ccnu.edu.cn

ABSTRACT
This paper discusses a parallel task-scheduling algorithm in network cluster computing system based on the model of multiprocessor parallel task scheduling. Compared with other methods, heuristic table scheduling can achieve better result with less cost. We designed this parallel task-scheduling algorithm for cluster system based on heuristic table scheduling algorithm. First, define the two properties of priority as: t-level (top-level) and b-level (bottom level). Then, Critical path is defined. Third, consider critical path node. Finally, we execute node-transferring scheduling. We apply a series of table scheduling algorithms, such as DLS (Dynamic level Scheduling, DLS) and MCP (Modified Critical Path) to compare performance. Our simulation experiment result shows this algorithm greatly improves scheduling performance.

Keywords: cluster computing; multiprocessor job scheduling; Critical path; Node-transferring; scheduling algorithms.

1. INTRODUCTION

At present, cluster is widely used in many systems, especially web service and application service systems. Various task-scheduling algorithms are developed on cluster systems. Task-scheduling cluster system is to allocate certain resources to each task and assign the start and stop time. The object of task scheduling is to minimize the time span, that is to say finishing the task as early as possible.

Typical, resources in a cluster system include processor, memory, RAID, multi-channel communication sub-system and so on. All of these resources can be seen as special processors. The execution and transition of tasks in system depends on many resources at the same time. For example, many execution of task need using processor, memory and communication channel simultaneously in cluster system. The task needs many processors running simultaneously. It's called parallel task. To make full use of resources in cluster system, we should use parallel task efficiently.

2. FORMALIZED DESCRIPTION

We assume that a group of parallel task is submitted to a cluster system. Every parallel task needs resources such as processor, memory, communication channel and execution time on cluster system [1]. In order to deal with the needs of given task, there are many node processors and many groups of processors running in the network cluster system.

Within the cluster system, different group of processors may have different execution time for the same parallel task. So, for every parallel task, the scheduling algorithm should decide the assignment of processor, then decide the execution time and minimize the time span to finish this work in cluster system.
This problem can be formularized as following: suppose that the network cluster system includes m processors $p_1, p_2, \cdots, p_m$, users submit n parallel task waiting for scheduling $s_1, s_2, \cdots, s_n$, every task have many choices for different processor group. For every processor group that can be chosen, (Q_{ij}, t_{ij}) shows the time the ith task need to use the jth group of processors [2]. So, the processor group model that can be used by every parallel task is described as follow:

$$Si = \{(Q_{i1}, t_{i1}), (Q_{i2}, t_{i2}), ., (Q_{ir}, t_{ir})\} \qquad (1)$$

Every Q_{ij} is a group of processor, that is, it is a subset of processor set $P = \{p_1, p_2, \cdots, p_m\}$; t_{ij} is the time that processor set used to execute the task. We need to give out a scheduling which can minimize the time span to finish the task. This is a typical problem of parallel task scheduling. We describe it as

$$Pm \mid set \mid T \min(m)(m \geq 2) \qquad (2)$$

This describing method includes three part of scheduling problem (using | to separate):

- The processors interconnected in the cluster system and processor number. For example P_m stands for m independent processors in the system.
- The constraint of scheduling instance. It includes constraint assignment of processor, priority between tasks, demanding for resources and so on. For example, set stands the task can have many choices to choose processor groups. The scheduling of processor can be preemptive or non-preemptive.
- The object of scheduling. For example Tmin represent the minimized time to finish the task so that throughput can be maximized.

* Supported by the Research foundation for the Nationality Emphases Laboratory (No.SKLSE04-018)

3. A PARALLEL TASK SCHEDULING ALGORITHM

Table scheduling algorithm includes static scheduling algorithm and dynamic table scheduling algorithm. The basic idea of dynamic table scheduling algorithm is calculating the priority of no scheduling node after every node assignment, and then arranges the node sequence according to the priority. These algorithms mainly have three steps:

A. Deciding the priority of no scheduling node;

B. Scheduling the node with highest priority;

C. Assigning this node to the processor that can be executed early.

Following those three steps, we can probably get a good scheduling algorithm [3]. The dynamic scheduling algorithm is normally more complex than the static scheduling algorithm. In this paper, we design a parallel task-scheduling algorithm based on heuristic table scheduling algorithm.

3.1. Related definition

In the cluster system, parallel program can be expressed by a directed acyclic graph (DAG). It can be defined as a quadruple:

$$G = (V, E, C, W)$$

In it, $V = \{v_i, i = 1 : n\}$ stands for set of task node.

E stands for the communication among nodes and the set of directed edge with priority.

C is the set of communication cost on directed edges, between two ends of E, which is a directed edge $c(n_i, n_j)$. $c(n_i, n_j) \in C, e_{ij} = (n_i, n_j) \in E$.

W is the set of calculating cost. $W(n_i) \in W$ is the running time of node $n_i \in v$.

Define the two properties of priority as: t-level (top-level) and b-level (bottom level).

As depicted in Figure 1: the label behind node is the calculating time of node and the label on the edge is the communicating cost.

- t-level

The t-level property of a node is the longest route length from the entrance node to t_i (include t_i). The route length means the sum of weights of every node and edge. The length of longest route to (in) can be calculated as follow:

$$\begin{cases} tl(n_1) = 0 \\ tl(n_j) = \max\{t_i(n_j) + w(n_j) + c(n_j, n_i)\}, n_j \in prod(n_i), i = 2, \cdots, v \end{cases} \quad (3)$$

$\Pr ed(n_j)$ is the set of node n_j 's father node, n_i is the entry node of DAG.

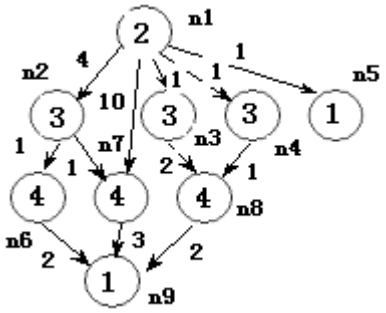


Fig. 1. A simple task graph

The t-level's property of node t_i is closely related to the beginning time. The beginning time can be defined after the certain processor is defined.

- b-level

The b-level property of a node is defined as the length of the max path from it to outlet-point. The b-level of a node is no more than the length of a DAG, and it can be concluded by the following formula:

$$\begin{cases} bl(n_i) = w(n_i) \\ bl(n_i) = \max\{bl(n_i) + w(n_i) + c(n_i, n_j)\}, n_i \in succ(n_j), i = 1, 2, \cdots, v \end{cases} \quad (4)$$

$succ(n_i)$ is the set of node n_i 's children node, n_v is the exit node of DAG.

- critical Path

Critical path is defined as the max length of a DAG, marked CP, and the length of its path is named as critical path length, marked CPL, and the node of its path is named CPN [4]. It is possible that there is more than one critical path in a DAG. The nodes of a critical path are named as critical path nodes, marked CPNs.

$$CPL = \max\{tl(n_i) + w(n_i)\} \quad (5)$$

- different times

The start time of node n_i on processor Q is marked $ST(n_i, Q)$ while the finish time is marked $FT(n_i, q)$, and the time to get data is marked $DAT(n_i, Q)$. If node have assigned to processor Q , then the start time $ST(n_i) = ST(n_i, Q)$, and finish time

$$FT(n_i) = FT(n_i, Q) = ST(n_i, Q) + w(n_i) \quad (6)$$

Time to get data

$$DAT(n_i, Q) = DAT(n_i, Q) = Max\{ST(n_j) + w(n_j) + c(n_j, n_i)\}.$$
$$n_j \in prod(n_i) \quad (7)$$

The scheduling length of task graph is the time parallel program running on the object system. That is

$$SL = \max\{FT(n_i)\} \quad (8)$$

3.2 Description of algorithm

In the process of scheduling, when two node is scheduled on one processor, it's weight of edge is 0, that is $c(n_i, n_j) = 0$, and the length of route is changed on route of this node. So t-level is variable. According to this principle, this algorithm (Critical directed path scheduling algorithm, CDSA) is divided into two steps: first step is preliminary scheduling, assigning the related task to one processor as more as it can be[5]; second step is migrating, migrating the task from main processor to neighbor processors with early start time and less communication cost[6].

Fist step:
In a DAG, the key route of CPN node is considered as main character, whose finish time decides final scheduling time. So in the processor of scheduling, node of CPN is first considered. For the start time, a node is decided by its father-node, so we should consider its father node before scheduling node of CPN.

The concrete algorithm is described as follow:
Initializing the preliminary scheduling sequence;
Let the entry node of key route be the first node of scheduling sequence;
Assume the next node of key route is next;
for processing every CPN node in key route
if all the father node of next is in preliminary scheduling sequence
then putting next into preliminary scheduling sequence;
let next CPN be next
else
for processing all the father node out of preliminary scheduling sequence
if existing maximum and early finish time
then its father node is CNP
else if existing maximum successor task
then assume its father node is CNP
else existing minimum task finish time
its father node is CNP
endif
endif
if all the father node of NP is in the preliminary scheduling sequence
then let NP in the preliminary scheduling sequence;
else using this method in recursion
endif
endfor
endif
endfor
for all other nodes
the last starting node in the preliminary scheduling sequence;
endfor

Second step:
After preliminary scheduling of DAG, we execute node-transferring scheduling of the result. CPN-Dominant Sequence decides the priority of migration of task. Algorithm of node migration is described as follow.
Algorithm:
Initialize processor linked list, the number of processor p=v
Initialize linked list of scheduled node
Produce the CPN-Dominant Sequence of critical path in DAG
Initialize a pointer (direct to the first node of CPN-Dominant Sequence)
for every node in CPN-Dominant Sequence
do
direct the pointer to n_x
let $ST(n_x) = tl(n_x)$
for every processor q_y in processor linked list
do calculating $DAT(n_x, q_y)$
if $DAT(n_x, q_y) \le ST(n_x)$
then migrating nx from the running processor q_x to processor q_y
if processor q_x is empty
then delete the processor q_x from linked list
endif
$ST(n_x) = ST(n_x, q_y)$
$FT(n_x) = ST(n_x) + w(n_x)$
Else n_x no migration
endif
endfor
endfor
$SL = \max\ i\{FT(n_i)\}$

4. ANALYSIS OF SCHEDULING ALGORITHM

Because the time of calculating node t-level is $O(e)$, the time complexity of preliminary scheduling for a certain reason is $O(e)$. The time complexity of producing CPN-Dominant Sequence is $O(e + v)$ 1 for v_2. There is v loops in first level and v "for" loops at worst in second level. In the second loop, the complexity time of $ST(n_x, q_x)$ is $O(e)$, then the time of first "for" loop is $O(ev2)$, so the complexity time for whole algorithm is $O(ev2)$ at worst.

We illustrate the DAG with this algorithm just like Figure.1. Table 1 is the t-level, b-level of node and the priority of node in CPN-Dominant Sequence.

Table 1. priority of node Sequence

Sequence	node	t-level	b-level
1	n1	0	23
2	n2	6	11
3	n7	12	8
4	n6	10	7
5	n3	3	12
6	n4	3	14
7	n8	7	8
8	n9	22	1
9	n5	3	1

The critical path of DAG is {n1, n7, n9}, and the priority sequence of critical node is {n1, n2, n7, n4, n3, n8, n6, n9, n5}.

The result of algorithm using on DAG is expressed in Figure.2, the number after node is start time.

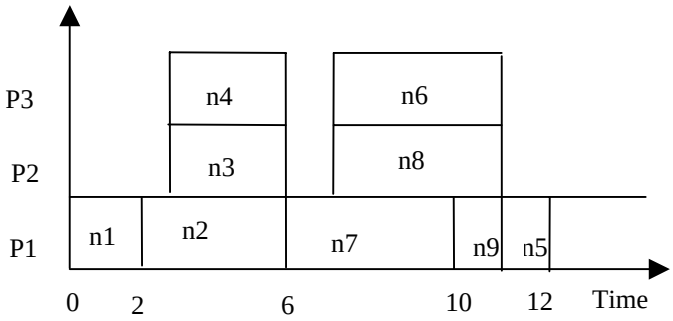


Fig. 2. The result of algorithm

5. EXPERIMENTAL RESULT

For the NP completeness of scheduling, we can't get the most optimal solution if using the heuristic algorithm. We need to use the arbitrary task graph, the bench-mesh task graph, the tree task graph and others in algorithm, and then compare performance of them. We do the emulating experiment in the environment of 100 nodes. The kilo mega exchangers connect these nodes. Routers connect all cluster system and external networks. Figure.3-5 shows the result of emulating experiment.

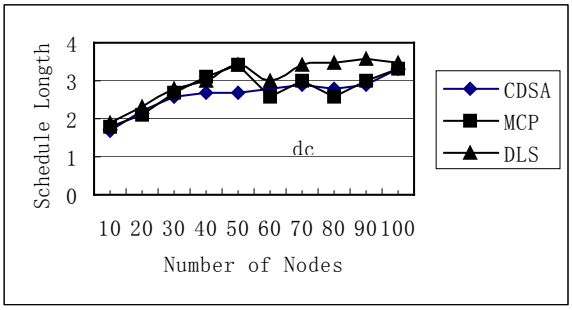


Fig. 3. The comparing result of positive tree graph

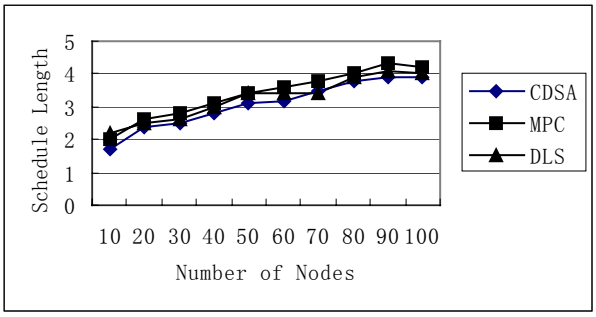


Fig. 4. The comparing result of multiple entries

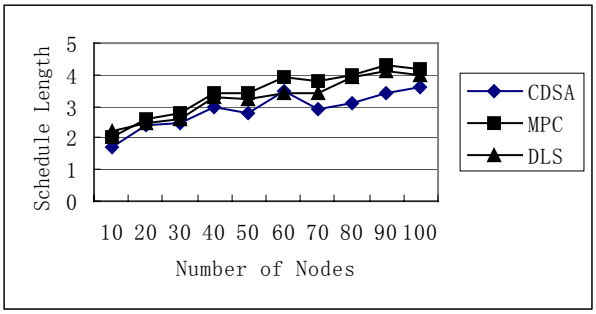


Fig. 5. The comparing result of multiple key routes

6. CONCLUSION

The performance of parallel task execution in network cluster system is mainly affected by scheduling algorithm. This paper presents an improved scheduling algorithm in network cluster system based on multi-processor parallel task model. According to our experiment results, this algorithm greatly improves the task scheduling performance without remarkably increasing complexity.

7. REFERENCES

[1] F.F.Zhou, Y.L.Xu, et al, "No-wait scheduling in single-hop multi-channel LANs", Information Processing Letters 93 (2005), 19–24.

[2] J.G.Huang, J.N.Chen, et al, "Parallel-job Scheduling on Cluster Computing Systems", CHINESE JOURNAL OF COMPUTERS, June 2004,765-771.

[3] B.Feng , J.Sun, "A Heuristic Algorithm for Task Allocation on Distributed Multiprocessor System", Computer Engineering, July 2004, 63-65.

[4] P.Yong, N.Gua, et al, "A Multiprocessor Taskscheduling Model for Berth Allocation: Heuristic and Worst-case analysis", Operations Research Letters 30(2002).343 – 350.

[5] S. S. Steven, Preemptive, "Multiprocessor Scheduling with Rejection", Theoretical Computer Science 262(2001), 437–458.

[6] T.C. Edwin, G.Chen, et al, "Semi-on-line Multiprocessor Scheduling with Given Total Processing time", Theoretical Computer Science 337(2005),134–146.

The Numerical Calculation Tested to Electronic Equipment Cabinet Random Vibrations

Yanping Liu, Xingxia Gao
Key Laboratory of Condition Monitoring and Control for Power Plant Equipment of Ministry of Education, North China Electric Power University, Beijing 102206, China
Email: lyp@ncepu.edu.cn

ABSTRACT

The dynamic characteristic and random vibration response characteristic of electronic equipment cabinet were simulated by the finite element analysis software— ANSYS according to the standard and request of mechanical structures for electronic equipment-Tests. First, 3-D finite element model of the cabinet was built. Next, cloud figures of deformation and stress of the model were obtained by loading and it could confirm the maximum of deformation and stress of the cabinet and probability of occurrence within the particular vibration frequency. Last, through more analysis of the frequency response curve, which could confirm harmful consequences random vibrations caused and weak link of the cabinet structure. The research results show that the simulation tests approximate the reality. It provides more convincing evidence for aseismic design and dynamic optimization design in cabinet structure.

Keywords: numerical calculation, electronic equipment cabinet, random vibration, finite element, ANSYS.

1. INTRODUCTION

Since our country having joined WTO, electronic products have entered into the international market, where if anyone wants to keep a firm status and possess strong competitiveness, the strict and reliable international standards of tests must be followed. However, in our country, random vibration-tests of electronic equipment mechanical structures, no uniform standard is issued and with the rapid development of economy in the modern society, customers pay more and more attention to the capability of bearing vibration of electronic equipment, especially that of bearing earthquake.

Mechanical force that the electronic equipment cabinet receives in the real working environment has various kinds of forms，such as vibration, impact, centrifugal force and frictional force produced by mechanism motion etc., Among which vibration and impact are the most harmful to the electronic equipment, they can cause two kinds of destruction as follows: (1) At some excitation frequency the equipment produces resonance, whose amplitude is more and more large, and is destroyed because of vibration acceleration beyond the limit one in the end or impact force beyond its intensity limit. (2) Stress caused by vibration acceleration or impact is well below the strength of material under the quiet load, but long-time shake or numerous impacts would make the equipment fatigue failure. Non-stationary vibration of the electronic equipment may take place under any of the following: (1) during transportation of products, (2) existing blasting earthquakes around the factory building probably, (3) earthquakes, (4) the running machines in the workrooms [1]. In the paper, through the numerical calculation, various kinds of response values of the cabinet in the environment of random vibration tests were analyzed and discussed.

2. REQUIRES AND COURSE OF RANDOM VIBRATION TESTS

Vibration test requirements:
1) The cabinet or rack configuration and loading condition for test set-up, shown in Fig.1.
2) The test response spectrum of the waveform shall match or exceed the required response spectrum.
3) During the test, it is necessary to measure the displacement of the upper side of the enclosure. The maximum deflection of the enclosure relative to the mounting surface point shall not over 50 mm.
Test conditions:
A cabinet or rack shall be mounted direct to the shaker table in accordance with the intended bolt-down positions and requirements in IEC 60068-2-47. The test shall be performed under the conditions of Fig.1.

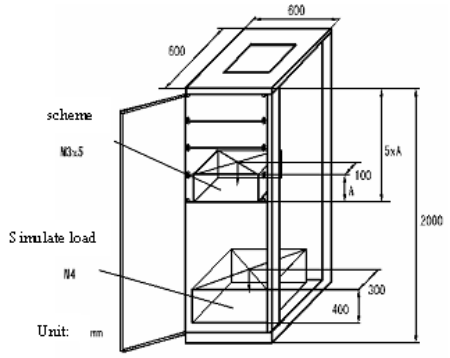


Fig. 1. Cabinet or Rack Configuration for Test Set-up

Test procedure:
1) The test wave for the seismic test shall be a synthesized waveform as described in Fig.2. Its zero period acceleration shall conform to the values of the severity levels in accordance with the requirement level.
2) Testing axis: each axis once—x, y, z. Changing once the directions should be began from test in advance.
3) Loading way: exerting the acceleration load on the single axis direction, the concrete values shown in Fig.2.
4) Duration: indicated in Fig.2.
Assessment following the test:
1) After testing, parts are not allowed to bring the deformation or destruction of influencing form, fit or function.
2) The earth-bond continuity shall be good in accordance

with 6.1 of IEC61587-1.

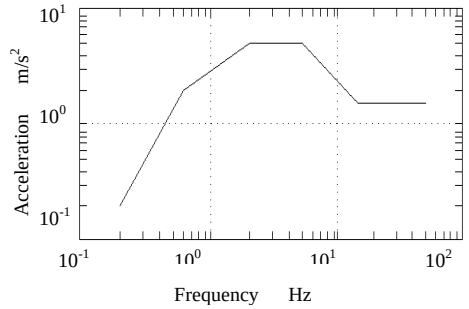


Fig. 2. Required Response Spectra

3. NUMERICAL CALCULATION PROCEDURE FOR THE RANDOM VIBRATION TEST

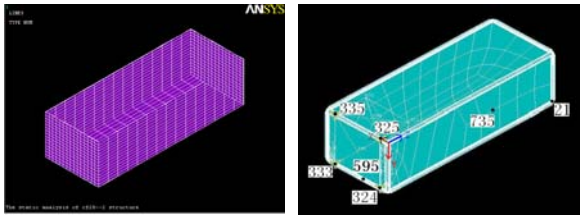
Numerical calculation of using the finite element software to perform the random vibration test is a spectrum analysis in substance. Spectrum analysis, an analysis technology that associates modal analysis result with a known spectrum, is mainly used for specifying the dynamic response of random or changing over time load, such as earthquake, wind, marine wave, engine vibration of the rocket etc. Three types of spectra are available for a spectrum analysis of ANSYS: Response Spectrum, Dynamic Design Analysis Method (DDAM) and Power Spectral Density (PSD). Response spectrum and DDAM analyses are deterministic (quantitative) analyses because both the input and output are actual maximum values. Random vibration analysis, on the other hand, is probabilistic (qualitative) in nature, because both input and output quantities represent only the probability that they take on certain values. This paper utilizes the large-scale finite element software ANSYS to numerically simulate and analyze random vibration (seismic) test, namely a PSD analysis, which consists of six main steps:

1) Build the model, including defining job name, analysis title, element type, element real constants, material properties, model geometry and so on, meshing units, and applying loads.
2) Obtain the modal solution.
3) Expand the modes.
4) Obtain the spectrum solution.
5) Combine the modes.
6) Review the results [2,3,4].

4. NUMERICAL CALCULATION INSTANCE OF THE RANDOM VIBRATION TEST FOR THE ELECTRONIC EQUIPMENT CABINET

4.1 Build the finite element model

The finite element model for PSD is the same to “The Numerical Simulation and Analyses Tested to Electronic Equipment Cabinet Structure Statics On The Basis of CAE Technology of ANSYS” [5]. As it point out, three-dimensional drawing application software PRO/E, UG, etc. may be utilized to set up entity's geometry model too. Then the geometry model is channeled into numerical analysis software ANSYS applying IGES file layout—a standard form which exchanges and shares model between different CAD and CAE system—or PARASOLID text form. And then model is repaired topologically or geometrically and simplified geometrically. The entity is established and its type is added and the net is divided. The advantage in modeling of the three-dimensional drawing software as PRO/E, etc. is easy to use, swift, to have such characteristics as parameter, dependence, and seriation and so on for the complicated model. While establishing models, it will bring you the facility while revising in the future. As long as change relevant size, it will become another part that you need. However, the cabinet structure roof beam in the article is mainly cold-curved and rolled by the sheet metal. Its structure is more special, but the model is sat up adopting PRO/E while channeled into ANSYS would produce a large number of form distortions. A large amount of topological repair was needed and it is very simple to build model directly by ANSYS, so the finite element model that using ANSYS to set up directly like Fig.3 (a).



(a) The Finite Element Model (b) Applying Constraints

Fig. 3. The Finite Element Model and Constraints of the Cabinet

4.2 Apply loads and obtain the solution

According to the standard of test and the cabinet material selected for use in design, in the paper, the material trade mark of the cabinet is Q235A, Elastic module of the material is $2.0*10^{11}$Pa, Major Poisson's ratios is 0.32. In modal analysis, four nodes (324,325,333,335) are established on the screw fixed bearing of cabinet corresponding finite element model and restrained completely, shown in Fig. 3(b). According to the frequency relation with acceleration illustrated in Fig.2, during the course of calculating response spectrum solution, the acceleration load is exerted at four screw bearings (nodes) along X, Y, Z axis separately, thus exerting of load is finished. The procedure of solving concretely is as follows:

1) Solve the mode shapes

Modal solution of the structure is a premise of spectrum solution, whose purpose is for the natural frequencies and mode shapes of a structure. When solving, remember these points: (1) mode extraction may adopt Subspace method, Block Lanczos method, Reduced (Householder) method. It is Block Lanczos method that was adopted in this paper. This solver performs well when the model consists of shells or a combination of shells and solids. The velocity is faster but requires about 50% more memory than Subspace; (2) the number of modes extracted should be enough to characterize the structure's response in the frequency range of interest. Mode number chooses 20 according to experiences; (3) the material damping must be specified in the modal analysis. This selects 0.05 as the standard of test; (4) it is sure to constrain those Degree of Freedom (DOF) where you want to apply a base excitation spectrum [6,7,8].

2) Expand the modes

Only expanded modes are used for the mode combination step. The mode expansion can be performed as a separate step, or can be included in the modal analysis phase by combining the modal solution and mode expansion steps by

including the MXPAND command. Expanding the modes applies not just reduced mode shapes from the reduced mode extraction method, but to full mode shapes from the other mode extraction methods (including Block Lanczos method) as well. Its purpose is to review mode shapes of structure of the cabinet in the postprocessor. This paper took mode expansion as a separate step and expanded all the modes. Before or after the expansion pass, leave SOLUTION with the FINISH command explicitly.

3) Obtain the spectrum solution

The followings are available for a random vibration analysis.

(1) Enter SOLUTION.

(2) Define the analysis type and analysis options.

(3) Specify load step options.

(4) Apply the PSD excitation at the desired nodes, and the UX, UY, and UZ labels imply the excitation direction.

(5) Begin participation factor calculations for the above PSD excitation.

(6) Specify the output controls.

(7) Start solution calculations.

During the course it showed the input PSD in Fig.4, which conformed to the required PSD.

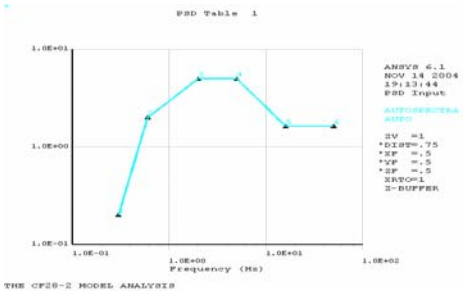


Fig .4. The Input PSD during the Spectrum Solution

In addition, similar to response spectrum analysis [9,10], a random vibration analysis may be single-point or multi-point. The test adopted single-point base excitation, it can only exert restraint which is exerted in modal analysis, namely at the 4 nodes on the screw bearing. In a single-point random vibration analysis, one PSD spectrum is specified at a set of points in the model, while in a multi-point random vibration analysis, different PSD spectra at different points.

4) Combine the modes

Mode combination mainly has two items: defining analysis type and choosing analysis type spectrum. Only the PSD mode combination method is valid in a random vibration analysis. This method triggers calculation of the one-sigma (1σ) displacements, stresses, etc. in the structure. The mode combination method determines how the structure's modal responses are to be combined, If select displacement as the response type, displacements and stresses are combined for each mode on the mode combination command; velocity, velocities and stress velocities are combined; acceleration, acceleration and stress acceleration also.

5. Analysis and post-processing

After finishing the above-mentioned solution, 1σ results of the finite element model of the cabinet can be observed through the general postprocessor (POST1). Cloud graphics as well as vector displays of deformation and stress of the cabinet when loaded at X, Y and Z direction, is shown in Fig.5, Fig.6a) and b), Fig.7a) and b), Fig.8a) and b). It draws the conclusion obviously from graphics that the deformation and stress of the electronic equipment cabinet located under the earthquake of certain frequency range. It makes for further analysis of the adverse consequences of seism and specifies the weak points of the cabinet structure. The time history preprocessor, POST26, can review the frequency response curve for displacement, velocity or acceleration of the finite element model. The paper takes the displacement response curves of node 595(maximum deformation), node 735(anyone at the middle of the cabinet), node 21(where the upper beams joint), Fig.9 for examples.

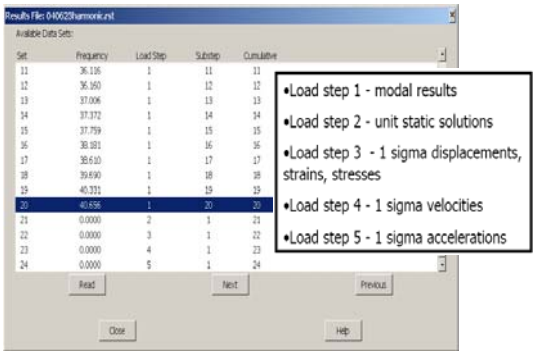
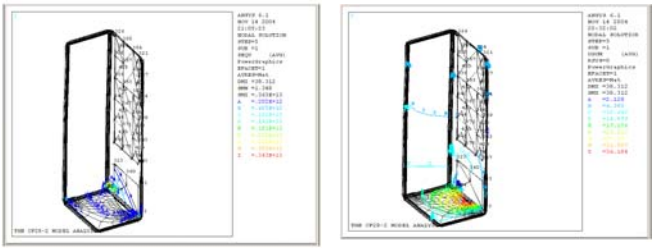


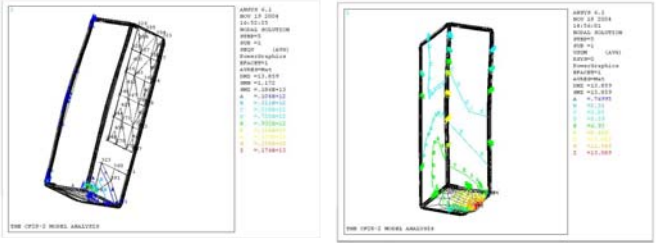
Fig. 5. 1σ Results of the Finite Element Model of the Cabinet

From Fig.5, 1σ results, namely load step 3, 4 or 5, are typically used for: first passage failure calculations, that is, the probability that the displacement at a DOF will exceed a displacement limit in a given time period; fatigue calculations, based on the premise that the stress level is at or below 1σ 68.3% of the time, between 1σ and 2σ 27.2% of the time (95.4-68.3), and between 2σ and 3σ 4.33% of the time (99.73-95.4), and above 3σ only 0.27% of the time (100-99.73).



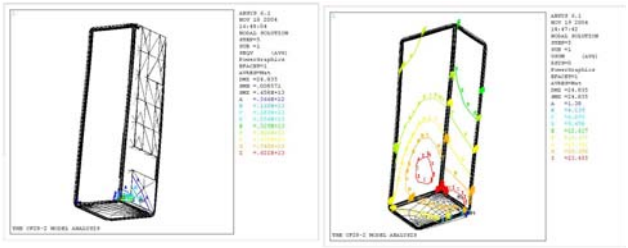
a) Stress Isopleths Graph b) Deformation Isopleths Graph

Fig.6 Loading at Z-direction



a) Stress Isopleths Graph b) Deformation Isopleths Graph

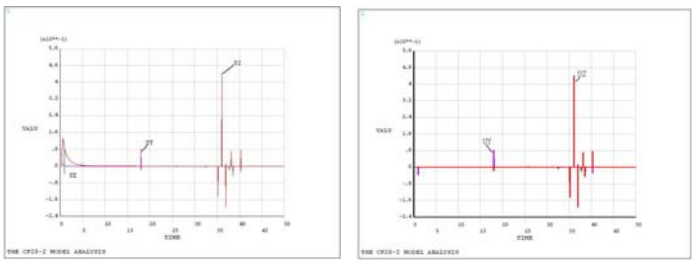
Fig. 7. Loading at X –direction



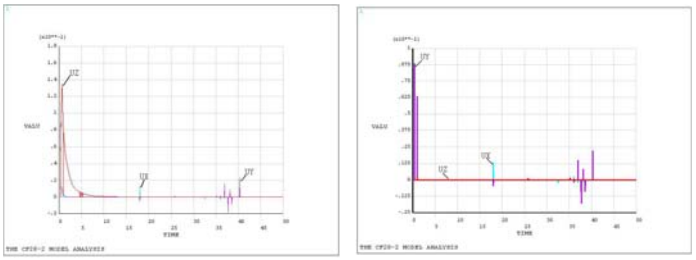
a) Stress Isopleths Graph b) Deformation Isopleths Graph

Fig.8 Loading at Y-direction

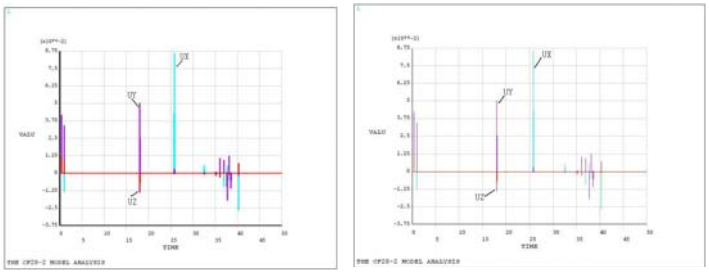
The nodes that the deformation and stress produce are found clearly from vector displays of deformation and stress. The node of the maximum stress is 324 and the maximum deformation is 595. By the time history preprocessor POST26 the displacement- frequency response curves of the nodes 595, 735 and 21 are shown in Fig.9a), b), c), d), e), f), g), h), i).



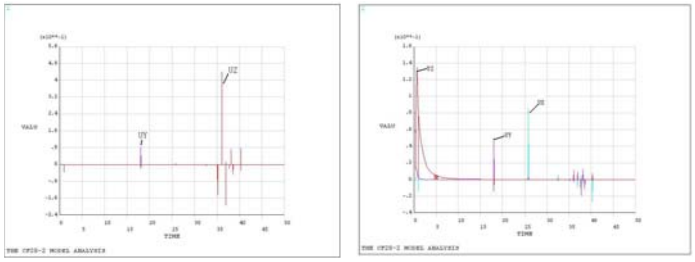
a) X Direction (595) b) Y Direction (595)



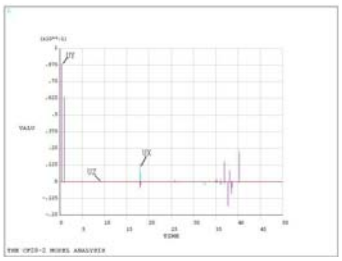
c) Z Direction (595) d) X Direction (735)



e) Y Direction (735) f) Z Direction (735)



g) X Direction (21) h) Y Direction (21)



i) Z Direction (21)

Fig. 9. Displacement-frequency Response Curves

6. Conclusions

(1) From deformation isopleths graph, when loading at X, Y or Z direction, 1σ displacement results are correspondingly 13.859mm, 24.835mm and 38.312mm, whose happening probability is 68.3%. The maximum deflection wasn't over 50mm required in the test. The deformation between 1σ and 2σ, namely 76.624mm, is more than that of requires of test standard only loading at Z direction, and its probability is 27.1%. The deformation over 3σ is 41.577mm、74.505mm、114.936mm at three directions, that is. Deformation at Y or Z direction is beyond that of requires of test standard, but its probability only 0.27%. And from above graphs the weak link may be visually found, which is in the junction of the base plate and the crossbeam of bottom, where should be strengthened in the design.

(2) From stress isopleths graph, the stress that locates in four screws where test jig connects with cabinet base is most dangerous. Therefore, the firmer connection way should be chosen while testing.

(3) From displacement-frequency response curve of three-direction separate loading, the displacement is the biggest in the middle part of the panel vertical to the loading direction. And the response parallel to the direction of the panel is far smaller than that perpendicular to its direction. Moreover, the encouragement panel or its junction with beams is most serious, which should be paid much attention to in the design.

(4) The numerical simulation of this paper can not only be simply identical with random vibration test, but draw the dangerous situation produced by random vibration qualitatively.

In sum, the random vibration simulation and analysis plays a very important role in improving aseismatic characteristic of products or aseismatic optimal design. It cannot merely contribute to the most optimal solution to seek at the stage of research and development of products, but obviously shorten product development cycle, reduce the production cost, guarantee product quality and thus increase economic efficiency.

7. References

[1] C.T. Qiu, "Design Principle of the Electronic Equipment Structure [M]", Nanjing: Southeast University Publishing House, 2005.

[2] X.B. He, C.B. Zhang, et al, "Dynamic Analysis of Electronic Cabinet in Vehicles [J]", Mechanical Engineering & Automation, 2006, (1): 46-48.

[3] The Agency of the American ANSYS in China, "The Guide for Dynamics Analysis with ANSYS [M]", Beijing,

2000.
[4] C.W. Wang, Y.A. Zhang, “Applications of Fatigue Analysis on Random Vibration for Fatigue Life Forecast of Airborne Equipment”, China Mechanical Engineering, 2004, 15(21): 1906-1908.
[5] L.D. Zhang, Y.P. Liu, et al, “The Numerical Simulation and Analysis Tested to Electronic Equipment Cabinet Structure Statics on the Basis of CAE Technology of ANSYS”, Department of Electrical Equipment, 2004, (10): 34-37.
[6] G.Q. Wang, “The Numeric Simulation Technique in Engineering and Its Practice in ANSYS”, Xian: Northwestern Polytechnic University Press, 2000.
[7] X.Q. Wang, G.J. He, “Review on simulation methods of random processes in engineering”, Building Science Research of Sichuan, 2005, 31(1): 26-29, 33.
[8] Z.X. Li, “Vibration Resistance Design of Electronic Equipment”, Electro-mechanical Engineering, 2002, 18(1): 51-55.
[9] J. Yao, Q.H. Yao, “A Simple Analysis Method for Random Vibration Response”, Chinese Journal of Applied Mechanics, 2002, 19(1): 103-105.
[10] Q.J. Chen, H.D. Wang, “Response Characteristic Analysis of Large Ship-lifter Tower Structure under Seismic Loading”, Journal of Vibration and Shock, 2005, 24(4): 52-55.

Parallel Computing Research On Clutter Modeling And Computer Simulation Of Pulse-Doppler Fuze

Lei Xu, SiLiang Wu, Hai Li, Wei Jiang
School of Information and Technology, Beijing Institute of Technology
Beijing, 100081, China
Email: thunder.xu@gmail.com

ABSTRACT

Clutter has a significant impact on the performance of airborne pulse-Doppler radars and hence is an important component of any radar model. The calculation of clutter of fuse essentially involves identifying the regions of the earth's surface, which contribute clutter, and dividing these regions into sufficiently small elements according to the range resolution. The radar range equation may then be applied to the elemental area, clutter return accumulated in the end. For this method computation time is essentially proportional to input parameters such as the platform altitude and the pulse repetition frequency (PRF). Because of the very high PRF of fuse in contrast with normal airborne radar, it takes such a long time to model the clutter that the simulation cannot meet the efficient demand in practice. A parallel algorithm of coarse granularity was proposed and by solving the problems of load imbalance, communication consumption and data storage, we achieve the simulation and reduce the computation time significantly.

Keywords: Fuse, Clutter Simulation, Parallel Computing, Load Balance, Granularity.

1. INTRODUCTION

The modeling of ground clutter in airborne radar systems is an essential component of assessing radar system performance. Radar clutter is the resultant of vector of all non-target scattering echoes in the area of antenna illuminating. Clutter echoes are more complex than target return signals, and cannot be defined by precise mathematical formula. In general, an airborne platform will exhibit motion relative to the earth's surface, and hence different regions will possess different resolved velocity components relative to the platform. Therefore, the net clutter contribution at the target range is also distributed over velocity, with the total, peak and velocity extent of clutter depending upon the combination of radar parameters, operational parameters, antenna radiation characteristics and terrain type. The electromagnetic scattering characteristics of a particular region of the earth's surface is a function primarily of the wavelength, the polarization, and the incidence and observation angles, and the surface scattering parameters, in particular, the surface roughness and composition to the depth of penetration. Therefore, airborne radar clutter is subject to both spatial and temporal variation and is difficult to model [1].

There are two kinds of methods to model the clutter. One is the statistical property of clutter simulation and the other is the video signal of clutter simulation. Generally, clutter has been modeled as a stochastic process, often by the empirical fitting of distributions to observed data. When the radar resolution is relatively low, the Gaussian clutter model can be used whereby its envelope has Rayleigh distribution. When radar resolution is relatively high and the pitching angle is very small, the clutter amplitude distribution can be shown through normal logarithm distribution. The characteristics of clutter as Weibull distribution and compound-k distribution can express clutter distribution more precisely [2].

Actually, when the real clutter data are rarely measured and when more precise clutter simulation data are needed we must turn to the other method. Usually the area of antenna illuminating can be divided into cell units. The magnitude of the samples in the radar pulse returned from each cell is calculated once per PRF using the radar range equation [3]. Then the samples are summed up to get the clutter data.

In this paper, the clutter of fuse is modeled. The missile often flies in the low altitude and the clutter has more effect on the fuse. The clutter echoes are diverse on the amplitude and phase because of the uneven surface of the ground and sea. It is hard to find some probability distribution fully describing the character of clutter. In order to model such clutter more precisely, we should model video signals of clutter with range ambiguity and Doppler stretch pulse by pulse. The PRF of fuse is so high that it can reach 1MHz contrast with several dozen kilohertz of normal airborne pulse-Doppler radars. Therefore the calculation time, which is essentially proportional to PRF, is enormous.

In this paper, we have proposed to use parallel computing to increase simulation speed and have solved several problems of load imbalance, communication consumption and data storage.

The remaining part of the paper is organized as follows. The simulation theory of clutter is described in Section 2. The parallel algorithm is described in Section 3. Some problems of parallel computing and their solutions are also discussed in this section. Section 4 covers the computing environment and result and conclusion is described in Section 5.

2. CLUTTER SIMULATION THEORY

The earth's surface, which contributes clutter, is divided into small cell units according to the range resolution. Since the area of antenna illuminating is very large when the missile is in the low altitude, the clutter is ambiguous in range. The range ambiguity needs to be considered in the simulation.

Define rectangular coordinates OXYZ as follows: assume the projection of the fuse antenna in the ground level for the origin O, and define the axis OX in the ground level whose direction is the same as the projection direction of the beam center in the ground level, and set the axis OZ vertical whose direction is upwards. As shown in Fig. 1.

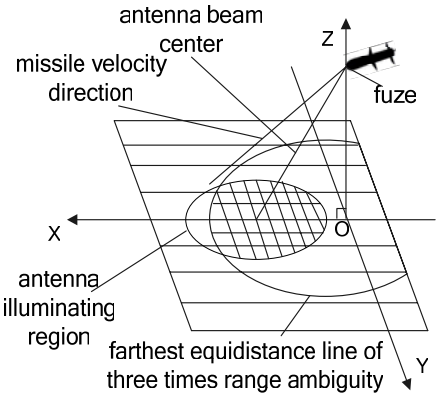


Fig. 1. Clutter region of fuse

Assume the antenna initial coordinates $(0,0,Z_{M0})$ and the missile velocity is V_M , and considers the angle between the direction of velocity and the OXY is θ_M , the angle between projection direction of the missile velocity in the ground level and the axis OX is φ_M . Then we can get the velocity direction as follows:

$$\begin{bmatrix} \mathbf{x}_M \\ \mathbf{y}_M \\ \mathbf{z}_M \end{bmatrix} = \begin{bmatrix} \cos\theta_M \cos\varphi_M \\ \cos\theta_M \sin\varphi_M \\ \sin\theta_M \end{bmatrix} \tag{1}$$

We can get the track of the missile as follows:

$$\begin{cases} x_M(t) = V_M t \cdot \mathbf{x}_M \\ y_M(t) = V_M t \cdot \mathbf{y}_M \\ z_M(t) = z_{M0} + V_M t \cdot \mathbf{z}_M \end{cases} \tag{2}$$

On the coordinates OXYZ, we divided the earth's surface into cell units, each unit as an equivalent scattering point. As the shadow region shown in Fig. 1., the region of antenna illuminating and the farthest equidistance line of three times range ambiguity determine the area of clutter covered. The shadow region is divided into many cell units with the step of ΔB , which is decided by the range resolution. The center coordinates for each cell units are defined as follows:

$$(x_m, y_n) = (m\Delta B + \frac{\Delta B}{2}, n\Delta B + \frac{\Delta B}{2}) \quad (m,n \in Z) \tag{3}$$

Each cell's Radar Cross Section (RCS) is

$$\sigma_m(x_m, y_n) = \Delta B^2 \times \sigma^0(x_m, y_n) \tag{4}$$

where $\sigma^0(x_m, y_n)$ is the scatting coefficient of point (x_m, y_n) in the ground level [4].

Assume that the scattering feature in the ground level is uniform, and then $\sigma^0(x_m, y_n)$ is only related to the incidence angle.

The amplitude of each scattering point is

$$A_{mn} = \frac{\sqrt{\sigma_m(x_m, y_n)} \cdot \sqrt{\rho_T(\theta_{mn})} \cdot \sqrt{\rho_R(\theta_{mn})}}{(x_m - x_M)^2 + (y_n - y_M)^2 + z_M{}^2} \tag{5}$$

Where $\rho_T(\theta_{mn})$ is antenna transmit gain and $\rho_R(\theta_{mn})$ is antenna receive gain. Because antenna pattern is for power, it needs square root of antenna gains to calculate the signal amplitude.

The time delay of each unit is

$$\tau_{mn} = 2\sqrt{(x_m - x_M)^2 + (y_n - y_M)^2 + z_M{}^2} \Big/ c \tag{6}$$

The intensity of IQ video clutter signal is

$$A_{\mathrm{Im}} = A_m \cos(-2\pi f_0 \tau_{mn}) \tag{7}$$

$$A_{Qm} = A_m \sin(-2\pi f_0 \tau_{mn}) \tag{8}$$

We can get the clutter video signals data by summing up all the echo signals from the cell units.

3. PARALLEL ALGORITHMS OF CLUTTER SIMULATION

In the serial algorithm, all echoes from clutter region need to be calculated in per PRF. The PRF is so high that the entire flight time of missile may include several millions of pulses. The clutter is modeled pulse by pulse and so the computation is intensive and time-consuming. To solve this problem, we have proposed parallel algorithms for clutter simulation.

When we turn to parallel algorithms, two factors must be considered:

1) Communication costs
2) Load imbalance cost

To get these two costs minimized is the central issue for designing an optimal parallel algorithm. For many applications, these two factors compete to waste cycles. For example, there is one parameter that can always be adjusted for performance: the granularity. Typically, when the granularity is small, one can get better load balance, but the communication cost increases. Decreasing granularity usually increases communication and decreases load imbalance. Increasing granularity usually decreases communication and increases load imbalance.

Minimizing the overheads of load imbalance and communication by choosing a proper granularity is the main goal in designing parallel algorithms.

In this paper, coarse granularity and fine granularity parallel algorithms will both be discussed.

Coarse granularity division means to divide the whole flight time into parallel parts to calculate and the minimum part of coarse granularity is one pulse.

Fine granularity division means to divide all the cell units into parallel parts to calculate and the minimum part of fine granularity are a cell unit.

We analyze both coarse granularity and fine granularity in the aspects of communication costs and load imbalance cost respectively.

3.1 Communication costs

When an aerial defense missile attacks a low altitude target, fuse transmits pulse signals all the time. The entire flight time is composed of several million pulses. All cell units in the clutter-covered region are calculated once per pulse. According to the demand, we take three times range ambiguity into account. The intermediate results that are calculated in one pulse need to accumulate the corresponding intermediate results in one, two and three times range ambiguity of previous three pulses. Then we will get the final result of one pulse and can store it in the file. As shown in Fig. 2.

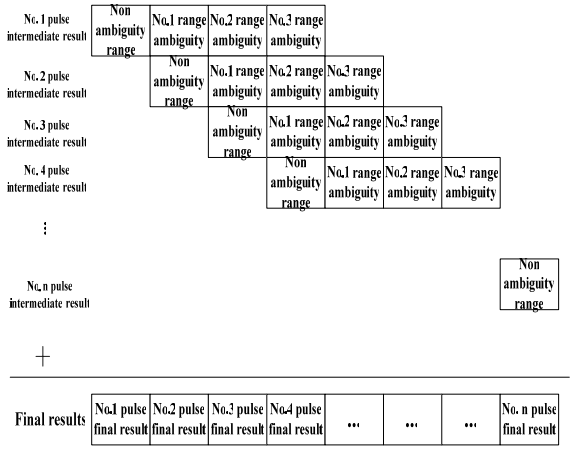


Fig. 2. The sum of results because of 3 times rang ambiguity

As we can see from Fig. 2., the final result of each pulse is the summation of the intermediate results of adjacent four pulses (except for the first three pulses). In the parallel algorithm, we need transfer these data among different nodes where a node represents a processor. This may cause communication costs. But it doesn't affect the final results when these data are transferred. This is like an embarrassingly parallel problem. We can collect results from all the nodes in the end. According to the demand, the file size of final results may reach 512MB. If we collect the results only once time in the end, great memory space will be needed to store the intermediate results. So we can collect the data every few pulses and store them in the file. The frequency of collection should be as few as possible because it is time consuming to do the synchronization before collection and to execute the collecting instruction. Coarse and fine granularity parallel algorithms have the same processing in the data transfer. How to communicate in this problem will have the same effects on both coarse and fine granularity.

3.2 Load Balance

Total calculation time of clutter simulation is

$$T = t_1 + t_2 + t_3 + \mathrm{L} + t_n + \mathrm{L} + t_N \tag{9}$$

where t_n is the calculation time of No. n pulse and

$$t_n = t_g \times n_g = t_g \times \frac{S_c}{S_g} \tag{10}$$

where t_g is the calculation time of each cell unit, n_g is the number of cells in one pulse, S_c is the area of clutter covered region and S_g is the area of each cell unit. Here, t_g is fixed and S_g is determined by the range resolution and is unchanged in the simulation. As we can see, t_n changes with S_c. S_c can be calculated as follows:

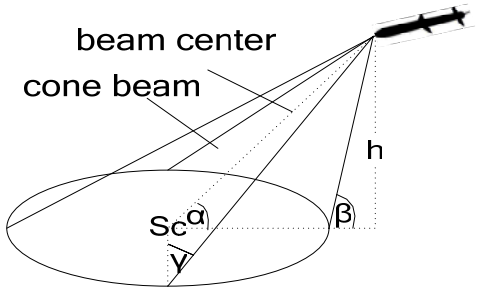


Fig. 3. Clutter-covered regions

Assume the simplest situation: cone beam is transmitted from antenna, illuminates the ground and forms an elliptical region. The angles α, β, γ are fixed, as shown in Fig. 3. and

$$S_c = \pi \times \left(\frac{h}{\tan\alpha} - \frac{h}{\tan\beta} \right) \times \left(\frac{h}{\sin\alpha \times \tan\gamma} \right)$$

$$= \pi \times \left(\frac{1}{\tan\alpha} - \frac{1}{\tan\beta} \right) \times \left(\frac{1}{\sin\alpha \times \tan\gamma} \right) \times h^2 \tag{11}$$

where h is the height of missile.

S_c in Esq. (11) is proportional to the square of h. So the calculation time of each pulse changes with the altitude of missile. With altitude reducing, the calculation time is becoming less. This will affect the load balance of coarse granularity.

For coarse granularity division, how to divide the entire flight time into parallel parts is the key problem. Because the pulse width is so short that the clutter region will not change for some time which may include many pulses. The calculation time is the same in this period of flight time. Thus each node can be assigned to calculate one pulse each time, which would ensure the best load balance for coarse granularity.

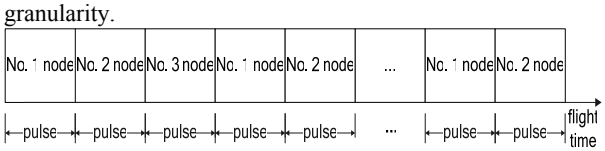


Fig. 4. Coarse granularity divisions

Fig. 4. shows how to assign one pulse to each node. The No. 1, 2, 3 in the figure means the serial number of nodes, where there are three nodes in parallel calculation assumptions.

For fine granularity division, as each cell unit calculation time is the same, as long as each node has the same number of cell units, it is easy to ensure load balance. However cell units in one pulse will be assigned to all nodes to do parallel calculation. The initial conditions such as missile location will be calculated more times than coarse granularity division in one pulse. For several millions of pulses, this will affect the whole calculation time.

Comparatively speaking, the coarse granularity division is better and is suitable for running in the cluster environment.

4. CLUSTER ENVIRONMENT AND EXPERIMENT RESULTS

Cluster computing environment is the mainstream technology of parallelism nowadays. It belongs to MIMD system. Every node of the cluster is an independent unit, which has its own operating system. The message passing SPMD programming model is used in the cluster. Every node runs the same copy of a program, but has different instruction and data flow. This kind of environment is suitable for coarse granularity parallel programs. In this paper, the hardware environment is an IBM's JS20 blade center server, with a total of six blades, each with double 1.6GHz PowerPC64 processors, 512kB L2 cache and 2G memories. Each blade is connected with one another through Gigabit Ethernet. This formed a typical cluster system. In the area of high performance computing, MPI has been the de facto standard for writing parallel applications [5]. The most popular MPI implementation MPICH is used. The operating system is SuSE Linux Enterprise Server 8.

The results of an example simulation are shown in Table

1., for a missile flight distance of 20 meters, a velocity of 3912m/s of the missile relative to the target, an altitude of 3 meters and a miss distance of 0.5m. The angle between the center of the antenna beam and the surface is 60^{o}. The PRF is 1MHz. When the PRF is determined, the total calculated time is inversely proportional to the area of cell on the same number of processors.

Table 1. Different cell size and different processors comparison

Area of cell	Numbers of processors	Time of calculation	Speedup	Parallel efficiency
$5\times5m^2$	1	203s	1	1
	2	105s	1.933	0.9665
	4	57s	3.561	0.8903
	8	32s	6.344	0.793
	12	24s	8.458	0.705
$2\times2m^2$	1	1263s	1	1
	6	211s	5.986	0.998
	8	160s	7.894	0.987
	12	110s	11.482	0.957
$1\times1m^2$	12	432s	/	/

Let $T_1(1,N)$ be the time required for the best serial algorithm to solve problem of size N on 1 processor and $T_P(P,N)$ be the time for a given parallel algorithm to solve the same problem of the same size N on P processors. Thus, speedup is defined as [6]

$$S(P,N)=\frac{T_1(1,N)}{T_P(P,N)} \tag{12}$$

Parallel efficiency is defined as [6]

$$E(P,N)=\frac{T(1,N)}{T(P,N)P}=\frac{S(P,N)}{P} \tag{13}$$

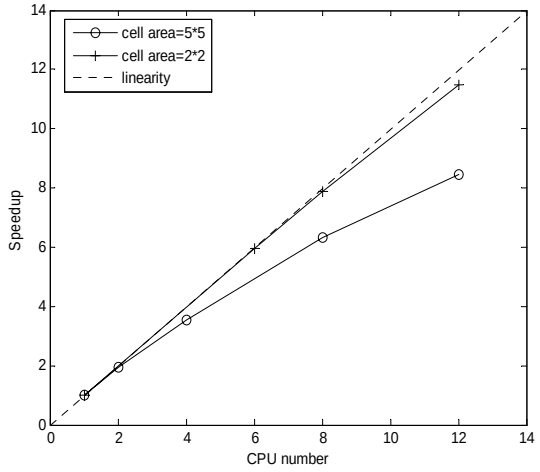


Fig. 5. Speedup of different size

As we can see from Table 1. and Fig. 5., satisfying speedup can be got in different size of the problem under the coarse granularity division. Comparatively speaking, when the size of calculation is larger, parallel efficiency will be better.

5. CONCLUSION

By analyzing the method of clutter simulation, we put forward a coarse granularity division parallel algorithm to solve the problem of computationally intensiveness. We have achieved the parallel algorithm under cluster hardware and MPI programming model environment. The results are satisfactory and the simulation time is reduced largely.

6. REFERENCES

[1] D.J. Browning, J.E Summers, “Computer modeling of ground clutter in airborne radar”, Radar System Modelling (Ref. No. 1998/459), IEE Colloquium on 8 Oct. 1998,6/1 - 6/6.

[2] X.C.Hu, X.D.Song, et al, “Clutter real time simulations of the airborne phased-array radar”, Radar, 2001 CIE International Conference on, Proceedings, 15-18 Oct. 2001,920 – 923.

[3] D. DiFilippo, G. Geling, et al, “Simulator for advanced fighter radar EPM development”, Radar, Sonar and Navigation, IEE Proceedings June 2001,148(3): 139 – 146.

[4] D. F. Boyer, G. L. Furjah, “Experimental Modified Reflectivity of Backscatter from the Sea Surface, at Very Low Grazing Angles, Compared to a Ducted Sea Clutter Model Used in Navy Radar Simulations”, IEEE Radar Conference, 2003.23-27.

[5] W. Gropp, E. Lusk, et al, Using MPI: Portable Parallel Programming with the Message Passing Interface, 2nd edition, MIT Press, Cambridge, MA, 1999.

[6] A. Grama, A. Gupta, Introduction to Parallel Computing, Second Edition, Addison-Wesley, Boston, MA, 2003.

A Novel Multilayer Gridless Detailed Routing Method

Mande Xie
College of Computer Science & Information Engineering, Zhejiang Gongshang Univ. HangZhou, Zhejiang 310035, P.R.China
Email:xiemd@mail.zjgsu.edu.cn

ABSTRACT

Based on a non-uniform-grid graph model, a self-adaptive iterative algorithm is presented for multi-layer gridless area routing. The algorithm can not only handle the case in which the number of metal layers is uncertain, but also greatly reduce the search space of multi-layer maze routing by converting multi-layer into multiple two-layer pairs. General speaking, the maze routing is slow. In order to accelerate checking the routing feasibility of a point in maze routing, an improved binary internal tree is introduced to manage routing obstacles, by which the comparison times are reduced.

Keywords: Non-uniform-grid, Gridless Routing, Binary Internal Tree, Detailed Routing, And Global Routing.

1. INTRODUCTION

The gridless routing method is firstly presented by Chen and Kuh [1]. They present a gridless channel router Glitter with variable width. Some improved gridless channel routing algorithms have been proposed in [2,3]. With the extensive employment of multi-layer routing technology, area-oriented routing methods gradually replace channel-oriented methods in order to effectively utilize routing resources. In general, there are two major approaches for the gridless area routing. One approach uses tile-based algorithm [4,5,6] in which routing region is partitioned into tiles by the boundaries of obstacles and the routing problem for searching a tile-to-tile path among these tiles managed by a corner-stitching data structure is simple. Searching a tile-to-tile path is quick due to the smaller number of tiles and the use of corner stitching data structure. However, the management of tiles is more complex, and a tile-to-tile path needs post-processing to obtain a final design rule correct route. Moreover, there are some difficulties in using a tile-based algorithm for multi-layer routing with more complex design rules. The other approach uses connection graph based [7,8,9] algorithm. At first, a connection graph is generated based on obstacles, and then the wire width and spacing information for nets is encoded into the graph. Then, a maze search algorithm is exploited on the graph to find the routing path. The work of [7] presents an implicit connection graph in which connection graph is not build in advance in order to accelerate the gridless detailed area routing. Based on the feature of gridless routing, the work of [9] employs an improved maze routing algorithm to get the most optimal solution if it exits. But the work of [7,8,9] is mainly done for two-layer routing.

Based on a connection graph, this paper presents a multi-layer gridless area routing algorithm, which exploits a self-adaptive iterative strategy. The algorithm can not only handle the case in which the number of metal layers is uncertain, but also reduce greatly the search space of multi-layers is uncertain, but also reduce greatly the search space of multi-layer maze routing by converting multi-layer into multiple two-layer pairs. For a multi-terminal net, the paper introduces a minimum Steiner Tree (SMT) method based on maximum dominant point concept to disassemble it into several two-terminal nets [11].

2. SELF-ADAPTIVE MULTI-LAYER GRIDLESS AREA ROUTING ALGORITHM

```
DetailRouting ()
{
      while( NetList is not empty)
     {
         1  SMTCenter();
         2  ModifySourceSinkBlk();
         3  while(the number of pins in the net > 2)
           {
         4    GenSMTNode(SMTCenterPoint);
         5    TwoPointMazeRouting(startPoint, SMTCenterPoint);
         6    TwoPointMazeRouting(endPoint, SMTCenterPoint);
           }
         7  if((the number of pins in the net == 1) &&
             (the flag of routing is true))
         8    AppendLineBlk(iNetIndex);
         9  else
           {
        10    TwoPointMazeRouting(startPoint, endPoint);
        11    AppendLineBlk(iNetIndex);
           }
     }
}
```

(a)

```
TwoPointMazeRouting（CSMPoint source,
                     CSMPoint sink)
Input: source-the source point
       sink-the sink point
{
     while(1)
   1 {  ProductTheRoutingLayerByIterator();
   2    ExpandBlock(iCurH, iCurV);
   3    GenNonuniformGraph();
   4    GenBlockGraph();
   5     if (routing(source, sink))
         {
   6         FindNewPath(source, sink);
   7         break;
         }
   8     if (last_try == true)
         {
   9         Record failed net seq;
  10         break;
         }
     }
}
```

(b)

Fig. 1. Multilayer gridless area-oriented routing algorithm

Maze algorithm is often employed in area routing. For two layer routing problem with reserved mode, routing grid is 2-dimension. For multi-layer routing problem, routing grid is multi-dimension, so maze algorithm must extend in multi-directions for multi-layer routing. As a result, time and space consumption are very tremendous.

Our self-adaptive multi-layer gridless area routing algorithm employs a special iterative strategy by which we convert a multi-layer routing into a sequence of two-layer routing. The algorithm can not only handle the case in which the number of metal layers is uncertain but also greatly reduce search space. For multiple nets, we employ the method described in [10](wire planning and reroute strategy) to overcome an influence of net order. The whole flow of the self-adaptive multilayer gridless area routing algorithm is described in Fig. 1(a), in which each net is handled one by one in the outer loop. For multi-terminal net, the algorithm disassembles it into several two-terminal nets by SMT method. For two-terminal net, it is handled directly. The multilayer maze routing algorithm for two-terminal net is described in Fig. 1(b). After finishing routing, the algorithm adds all the segments in the routing path into the set of obstacles. In the multilayer maze routing flow for two-terminal net described in Fig. 1(b), *ProductTheRoutingLayerByIterator* in Line 1 generates the current H-V routing layer pair. *ExpandBlock* in Line 2 expands all obstacles in the current H-V routing layer pair (not all layer) by the method described in session 2 and gets a set of expanded obstacles. By the definition 1, *GenNonuniformGridGraph* in Line 3 generate the NUG graph induced by the set of expanded obstacles, the set of net terminals and routing edges. *GenBlockGraph* in Line 4 marks and encodes obstacles type for every point in NUG graph, according to obstacle information. After the execution of Line 1-4 above, resolution space for current maze algorithm is formed. *Routing* in Line 5 finds a point-to-point routing path for two-terminal net on the NUG graph by the improved maze routing algorithm. If routing is successful, *FindNewPath* in Line 6 traces a routing path and the loop exits at once in Line 7, which ensures the number of routing layer required is minimum. Or the algorithm checks whether it is the last iteration. If so, the program goes to Line 1 and goes on the loop, otherwise sets the failure flag and exits the loop.

In following sessions, several important technologies used in this algorithm are addressed: self-adaptive iteration strategy, improved binary internal tree

2.1 Self-Adaptive Iteration Strategy

The algorithm handles multilayer routing problem by a self-adaptive iterative strategy. For reserved mode routing, *start_layer_type* is a flag, which indicates the current reserved mode is HVH… or VHV… *curr_h_layer* records the current horizontal routing layer and *with_up_layer_or_down* indicates the current vertical routing layer which forms a H-V routing layer pair with the horizontal layer indicated by *curr_h_layer*. *ProductTheRoutingLayerByIterator* in line 1 in Fig. 1(b), in which the multilayer maze routing algorithm for two-terminal net is shown, is fined in Fig.2. and the self-adaptive iteration strategy is also summarized in Fig.2.

For a routing instance with 4 metal layers in HVHV reserved mode, the running process is followed: the first iteration in Line3-4 generates the current H-V routing layer pair, 1-2 layers; the second iteration in Line 12-13 generates the current H-V routing layer pair, 3-2 layers; the third iteration in Line 9-10 generates the current H-V routing layer pair, 3-4 layers and sets *last_try* in Line16. For a routing instance with 5 metal layers in VHVHV reserved mode, the running process is followed: the first iteration in Line5-6 generates the current H-V routing layer pair, 2-1 layers; the second iteration in Line9-10 generates the current H-V routing layer, 2-3 layers; the third iteration in Line12-13 generates the current H-V routing layer pair, 4-3 layers; the fourth iteration

```
      ……
1   if(first_iteration is true){
2       if(start_layer_type == HVH){
3           curr_h_layer = 1;
4           with_up_layer_or_down = combined_with_next_layer;
        }
        else{
5           curr_h_layer = 2;
6           with_up_layer_or_down = combined_with_previous_layer;
        }
7       first_iteration = false;
    }
    else{
8       if(with_up_layer_or_down == combined_with_previous_layer){
9           if(number_of_layers > curr_h_layer )
10          with_up_layer_or_down = combined_with_next_layer;
        }
        else{
11          if(number_of_layers > curr_h_layer + 1){
12              with_up_layer_or_down = combined_with_previous_layer;
13              curr_h_layer += 2;
            }
            else{
14              break;
            }
     }
15  if(((curr_h_layer) == number_of_layers) ||
      (((curr_h_layer+1) == number_of_layers)&&with_up_layer_or_down ==
        combined_with_next_layer;))
16      last_try = true;
       ……
```

Fig. 2. The iterative strategy

in Line 9-10 confirms the current H-V routing layer pair is 4-5 layers, and sets last_try in Line 16.

2.2 Improved Binary Internal Tree

For the convenience of discussion later, some important definitions are firstly given.

Definition2 routing feasibility of point (RFP): if a point in the non-uniform-grid graph doesn't fall into any rectangle, which represents an obstacle, the point is feasible for routing. Otherwise the point is infeasible for routing.

In the maze routing (described in line 5 of the Fig.1 (b)), checking RFP is a very key operation. The reason is followed:

(1) Checking RFP at least requires to 4 comparison operations

(2) Given a set of obstacles $\tilde{R}^n = \left\{ \tilde{r}_1, \tilde{r}_2, \wedge, \tilde{r}_n \right\}$, where n is the number of obstacles, and then the complexity of checking RFP is $O(n)$. With the increment of n, the number of comparison operation obviously increases to check RFP.

(3) Checking RFP is a high frequent event in the maze routing, so it need particularly be accelerated.

In order to reduce unnecessary comparison operations and run time of the algorithm, an efficient data structure to manage obstacles is needed.

The data structure for layout has generally corner-stitching, bin structure, quad-tree and so on. Corner stitching is suitable for checking adjacent relationship among areas, but it is unsuitable for checking whether a point falls into a rectangle. Because bin structure depends on the size of partitioned grids, improper grid size results in space waste and repetitive comparison. Rectangles can be efficiently distinguished from different areas by the hiberarchy of quad-tree. But because each node in quad-tree stores the graphics, which are intersected with a horizontal bisector and a vertical bisector in corresponding area, many repetitive comparisons are

This paper is sponsored by Nature Science Foundation of Zhejiang Province(Grant Y105599)

generated. In addition, most additional operations are needed to maintain the data structure. In our algorithm, Improved binary internal tree is introduced to reduce unnecessary comparison and additional maintenance cost.

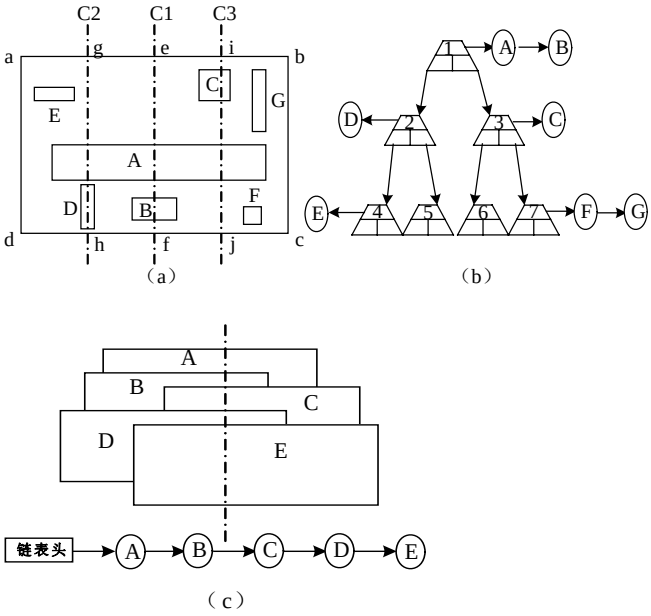


Fig. 3.（a）A partition （b）Corresponding interval tree （c） A special partition

When binary internal tree is constructed, along the special direction (horizontal or vertical), the routing region is recursively partitioned with a sequence of bisectors. In Fig.3 (a), a partition process along vertical direction is showed. In the figure, *abcd* indicates routing region; *A, B, C, D, E, F, G* indicates expanded obstacles. In the first partition, the whole routing region is partitioned into two parts with bisector c_1 and child rectangles *aefd* and *ebcf* are generated. The object of the second partition is areas *aefd* and *ebcf*. These areas (*aefd* and *ebcf*) are partitioned by bisector c_2 and c_3 respectively. As a result, four rectangle areas are generated: *aghd*, *gefh*, *eijf* and *ibcj*. And so on and so forth, each rectangle area which is generated in before partition is partitioned into two equal parts until the span of the sub-region in x direction is below a threshold value. Because the spans of all rectangle sub-areas in y direction are equal to the span of the whole routing region in y direction, an internal in x direction can represent a sub-area. The partition method along horizontal direction is similar to the above-mentioned method except that the cut line becomes horizontal. So the directions of all internals are y direction. With routing region partitioned, adding pointers to each node, which represents an internal, generates a binary internal tree.

Each node in a binary internal tree represents an internal [*start, end*], whose direction is perpendicular to the direction of cut lines. In fig.3.(a), the direction of internal is x direction. Each node stores all obstacles that are not only contained by this internal but also intersected with the bisector of this internal. Each non-leaf node has a right child node and a left child node, which recursively point to sub-internal [*start, (start+end)/2*] and [*(start+end)/2, end*] respectively. In Fig. 3(b), a binary internal tree corresponding to Fig.3 (a) is showed. The node 1, which is root node, represents the whole region area and the node 2,3 represent the internal [*d, f*] and [*f, c*] respectively. By the rule that each node stores all obstacles that are not only contained by this internal but also intersected with the bisector of this internal, obstacles *A, B, D, C, E, F, G* are stored in the node 1, 2, 3, 4, 7 respectively. The whole binary internal tree is shown in fig.3. (b). Now, while checking RFP, the followed step is done:

(1) Find the path from the root node to the node, which represents the internal that this point lies in. The node set N_{path} is generated (each node in this set lies in the path).

(2) Check whether this point fall into any obstacle lied in the obstacle list of node $N_i\,(N_i \in N_{path})$

So the complexity of checking RFP is reduced from $O(n)$ to $O(\log(w)+s)$, where *s* is the number of obstacles necessarily compared and *w* is the region width. Because of $s << n$, the run speed of the algorithm is obviously improved.

3. EXPERIMENT RESULTS

We implement our algorithm in C++ and debug it successfully. Lots of examples show our algorithm can not only handle the case in which the number of metal layers is uncertain, but also greatly reduce the search space and improve the quality of routing. In Fig.4, a small-scale routing example with three layers (HVH) is showed. In Fig.5, the routing result of a large-scale example (VHVH) is showed.

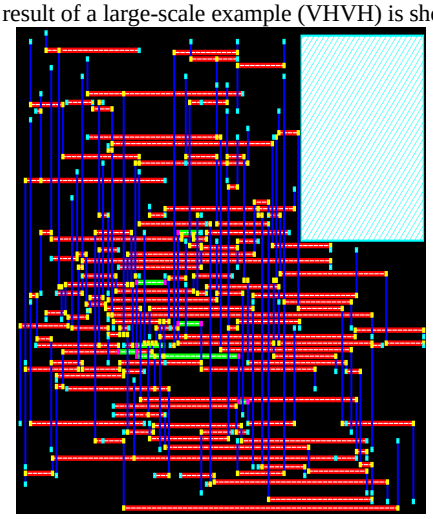


Fig. 4. A three layers routing instance in HVH reserved mode

The experiment data shows all instances have 100% routing rate and our algorithm has the ability to get around macro-obstacle automatically. Because a whole routing syst-

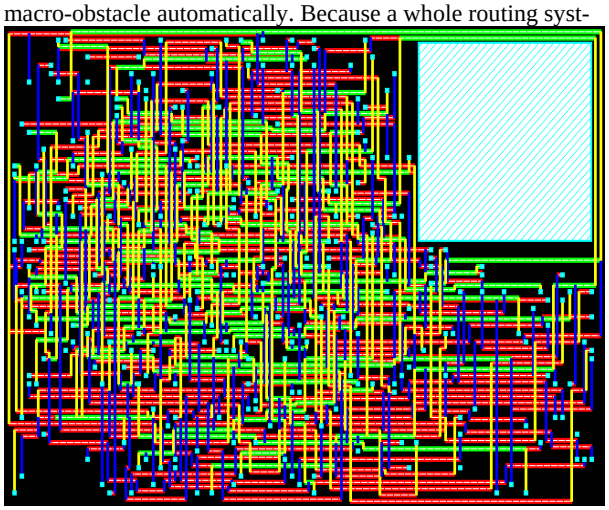


Fig. 5. A four layers routing instance in VHVH reserved mode

em(including global routing, crosspoint assignment) isn't finished, this algorithm isn't be tested by general benchmarks. To verify further this algorithm valid, we test this algorithm by random nets. The test program makes sure that each random net have at most 5 pins. In table1 several routing examples and corresponding routing results are given. In table1, several routing results are compared: 1) The same net number with different reserved mode and different routing layer; 2) the routing with accelerating strategy and without accelerating strategy. The experiment data shows our algorithm is valid and effective.

Table 1. Routing statistical result

Routing examples				Routing results			
Nets	Layers	Mode	Pins	Run times(s) without accelerating	Run times (s) with accelerating	Rates(%)	Length
20	3	HVH	68	1.1	0.98	100	11672
	3	VHV	68	1.3	1.1	100	12316
	4	HVHV	76	1.9	1.6	100	11716
	4	VHVH	71	1.5	1.3	100	11806
40	3	HVH	144	12.0	9.7	100	26312
	3	VHV	138	10.2	8.6	100	23960
	4	HVHV	140	14.4	11.7	100	26536
	4	VHVH	145	13.2	11.5	100	24464
60	3	HVH	212	51.4	42.9	100	33468
	3	VHV	207	45.4	39.1	100	32610
	4	HVHV	225	70.3	58.1	100	32680
	4	VHVH	202	39.5	33.5	100	29816
80	3	HVH	290	110.9	90.2	100	29372
	3	VHV	275	96.6	78.5	100	33180
	4	HVHV	297	149.3	125.5	100	32370
	4	VHVH	290	130.1	108.4	100	32322
100	3	HVH	356	185.2	150.6	100	34498
	3	VHV	359	196.2	160.0	100	36832
	4	HVHV	351	221.8	183.3	100	32948
	4	VHVH	346	208.1	172.1	100	36224
200	3	HVH	688	683.4	542.4	100	45836
	3	VHV	712	819.8	635.5	100	41496
	4	HVHV	677	921.5	698.1	100	43320
	4	VHVH	674	879.9	676.7	100	45118

4. CONCLUSION

Based on NUG graph mode, this paper presents and implements a multi-layer gridless area routing algorithm. The algorithm presents a self-adaptive iterative algorithm. For a multi-terminal net, the paper employs a minimum Steiner Tree (SMT) method based on maximum dominant point concept to disassemble it into several two-terminal nets [11]. The experiment data shows the algorithm is valid and effective

5. REFERENCE

[1] H.H.Chen, S.K.Ernest, a Gridless Variable-Width Channel Router.IEEE Transaction on Computer-Aided Design vol.CAD-5,No.4,October 1986.

[2] P.Groeneveld, A Multiple Layer Contour-Based Gridless Channel Router. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , Volume: 9 , Issue: 12 , Dec. 1990.

[3] R.Yazdani, M.R.Zargham, A Gridless Multilayer Channel Router. COMPSAC 90. Proceedings., 14th Annual International , 31 Oct.-2 Nov. 1990 Pages:363 –368.

[4] M.Sato, J.Sakanaka, et al, A fast line-search method based on a tile plane. in IEEE international symposium on circuits and systems,pp.588-591,may 1987.

[5] A.Margarino, A.Romano, et.al., A tile-expansion router, IEEE Trans. Computer-Aided Design, vol.CAD-6,pp.507-517,Jul 1987

[6] L.C.Liu, H.P.Tseng, et al, Chip-level area routing, in Porc. Interational Symposium on Physical Design, pp.197-204, Apr 1998.

[7] J.Cong, J.Fang, et al, An Implicit Connection Graph Maze Routing Algorithm for ECO Routing. ACM International Conference on , 7-11 Nov. 1999 Pages:163–167

[8] S.Q.Zheng, J.S.Lim, et al, Finding obstacle-avoiding shortest paths using implicit connection graphs, IEEE Trans. Computer-Aided Design,vol.15,pp.103-110,Jan 1996.

[9] S.J.Wang, H.T.Ge, et al, Research on VLSI Gridless Routing Algorithm, Journal of Circuit and System, 2002**.** 7(4) **.**

[10] J.Cong, J.Fang, et al, DUNE:A Multi-Layer Gridless Routing System with Wire Planning. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , Volume: 20 , Issue: 5 , May 2001 Pages:633 – 647.

[11] M.D.Xie, Q.Ma, et al, A Multi-layer Gridless Area Routing Algorithm Based on Non-Uniform-Grid Graph. 2005 International conference on communication, circuits and systems proceedings. pp 1258-1262, May,2005.

This paper is sponsored by Nature Science Foundation of Zhejiang Province(Grant Y105599)

The Analysis of Data Parallel Problem Based on Timed Petri Nets*

Xianwen Fang1, Zhicai Xu2, ZhixiangYin1
1Maths and Physics Department, Anhui University of Science and Technology,
Huainan, 232001, China;
Email: xwenfang@tom.com
2Chuzhou University, Anhui Chuzhou, 239012, China,
Email: xwliu@aust.edu.cn

ABSTRACT

As an intuitionistic graph modeling tool and a formal model with abundant mathematic knowledge, Petri nets are applicable for depicting the system characteristic of being concurrent, asynchronous and distributed. Data parallel means that the same operation acts on different data simultaneously. By using the Timed Transition Petri Nets (TTPN) to analyze the data parallel problem, the structures of the same operation can be found out, which is beneficial to the simulation of the data parallel problem in parallel machine. A Matrix Multiplication algorithm is simulated and realized in Dawning 2000. The result shows that the TTPN simulation of data parallel program is effective.

Keywords: Timed Transition Petri Nets (TTPN), Data parallel, Process, Simulation.

1. INTRODUCTION

At present, two kind of most important parallel programming models are the data parallel and the message passing. The programming rank of data parallel model is quite high, so programming is relatively simple, but it is only suitable for the data parallel question. The programming rank of message passing model is relative lower, but the message passing programming model may have the more widespread application scope, including data parallel and task parallel question.

The data parallel is that the same operation acts on the different data simultaneously, therefore it suits in the SIMD and the SPMD parallel machine. It also explained that the data parallel may effectively solve a kind of science and the engineering calculation question through the data parallel solution question practice in the vector machine, but regarding non- data parallel kind of question, it is very difficult to obtain a higher efficiency if it is solved through the data parallel way, the data parallel is not easy to express even unable to express other forms the parallel characteristic. The data parallel has not very effective compiler presently. Has the highly effective compiler, the data parallel procedure will be able to obtain the high efficiency in the sharing memory and in the distributional memory parallel machine, like this it can enhance the parallel programming efficiency and the parallel procedure transplant ability, further promote the parallel programming technology.

* Supported by Natural Science Foundation of China (30570431), Youth Foundation of Educational government of Anhui Province of China (2006jq1077)
**Corresponding author. Tel: 13155453020;

The message passing model is that between each concurrent execution process of the parallel program exchanges the information, coordinates the step, controls the execution through the message passing, the message passing generally faces the distributed memory, but it also suitable to sharing memory parallel machine. The message passing has provided the more nimble control method and the parallel expression method for the programmer, some data parallel program may be realized using the message passing model, but it is very difficult with the data parallel method.

At present, because the data parallel programming model lacks the highly effective compiler support, the data parallel question is mainly realized using the message-passing model. We use the timed transition Petri nets (TTPN) to analyze data parallel program. If we use the basic statement transformation rules in the literature [5,7,8], we will have an extremely complex model, which is disadvantageous to the analysis, because has the same operation in the data parallel program simultaneously to affect to the different data. According to the literature [3,4,6], we propose a new analysis method; it is benefit to deal with the data parallel program. Finally, we carry out the two 400×400 matrices multiplication in the Dawning 2000 parallel machines by the TTPN analysis method.

2. BASED CONCEPTION

Here, we only introduce several conceptions correlating with the paper close, other Petri Nets terms in the literature [1,2].

Definition1 [1]. A triple N=(P, T; F) is called a net iff

1) $P \cup T \neq \phi$
2) $P \cap T = \phi$
3) $F \subseteq ((P \times T) \cup (T \times P))$
4) $dom(F) \cup cod(F) = P \cup T$

$here, dom(F) = \{x \in P \cup T \mid \exists y \in P \cup T : (x, y) \in F\}$

$cod(F) = \{x \in P \cup T \mid \exists y \in P \cup T : (y, x) \in F\}$

Definition 2 [1]. Let N is a net, for $x \in P \cup T$,

$$^{\bullet}x = \{y \mid y \in P \cup T \wedge (y, x) \in F\},$$
$$x^{\bullet} = \{y \mid y \in P \cup T \wedge (x, y) \in F\}$$

$^{\bullet}x$ is called the preset of x, $x^{\bullet}$ is called the postset of x.

Definition 3 [1]. A 6-tuple N=(P, T; F, K, W, M) is called a place/transition net (P/T-net) iff

(1) $(P, T; F)$ is a net;

K: $P \to N_0$ (N_0 is natural number set), gives a (possibly unlimited) capacity for each place;

W: $F \to N_0 - \{0\}$, attaches a weight to each of the net;

M: $P \to N_0$ is the initial marking, respecting the

capacities. i.e. $\forall p \in P$: $M(p) \le K(p)$

(2)A transition $t \in T$ is enabled iff

$\forall p \in {}^\bullet t$: $M(p) \ge W(p, t)$

$\forall p \in t^\bullet - {}^\bullet t$: $M(p) + W(t, p) \le K(p)$

Definition4 [2]. A 6-tuple $TTPN = (P, T, F; \tau, W, M)$ is called timed transition Petri net (TTPN), iff

$P = \{p_1, p_2, \cdots, p_n\} (n \ge 0)$ is a finite set of place;

$T = \{t_1, t_2, \cdots, t_m\} (m \ge 0)$ is a finite set of transition, and $P \cap T = \Phi$;

$F \subseteq (P \times T) \cup (T \times P)$ is a directed arcs set;

$\tau : T \to 0 \cup Q^+$ (Q^+ is positive rational number)is time mapping function;

W: $F \to \{1,2,3,\ldots\}$ is called weight function;

$M : P \to I$ ($I = \{0,1,2,\cdots\}$)is net marking.

The transition enabled rules of timed transition Petri net is similar to the P/T net, but in the TTPN, the mark M enabled t firing, and the transition t can fire completely after τ time.

Definition 5. A timed transition Petri net TTPN= $(P, T, F; \tau, W, M)$, a transition $t \in T$ is enabled iff

$\forall p \in {}^\bullet t$: $M(p) \ge W(p, t)$

If $M[t > M'$ (a enabled transition t may yield a follower marking M' of M),for $\forall p \in P$:

$$M'(p) = \begin{cases} M(p) - W(p, t) & \text{当 } p \in {}^\bullet t - t^\bullet \\ M(p) + W(t, p) & \text{当 } p \in t^\bullet - {}^\bullet t \\ M(p) - W(p, t) + W(t, p) & \text{当 } p \in {}^\bullet t \cap t^\bullet \\ M(p) & \text{当 } p \notin {}^\bullet t \cup t^\bullet \end{cases}$$

3. THE TTPN SIMULATION OF DATA PARALLEL PROGRAM

In order to analyze the data parallel program with TTPN, parallel programming model realization based on the message passing is most important, then the source code is divided into many processes according to the procedure characteristic, using the timed transition Petri net to simulate each process to obtain corresponding the sub model, connecting each sub model by the process correspondence, then obtaining the TTPN model of data parallel program. Regarding each process, we may divide into it many segments, The j-th segment of process is model by the subnet in Fig.1 [6], where transition $S(P_i, j)$ represents the computation carried out by the segment, place $IN(P_i, j)$ models its input message buffer, and the weight K of arc connecting these two places corresponds to the number of messages that the segment must receive before its computation can start. If the segment does not require any message, place $IN(P_i, j)$ as well as the above arc will not be present. Place $C(P_i, j)$ is marked with a token when the control is passed to the segment, while place $C(P_i, j+1)$ is marked when the control is passed to the next one. Each subnet $Out(P_i, j, x) - Snd(P_i, j, x) - In(P_w, m)$ represent s the communication involving the message sent to segment *L* of process.

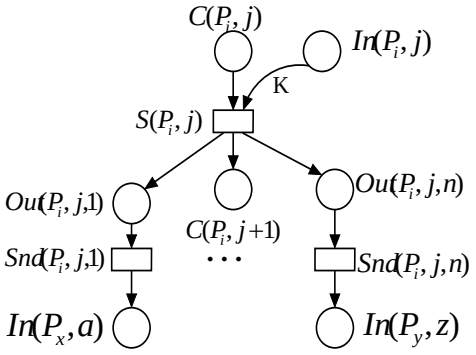


Fig. 1. The TTPN model of a segment in logical process

Loops whose body contains one or more segments are modeled as depicted in Fig. 2 [6], provided that the number of iterations *X* is known a priori, i.e. all the loops are of the "for" type. When the process computation reaches the loop, place $C(P_i, j)$ is marked with *X* tokens. Each time an iteration is completed; a token is added to place *Ndone*, transition *End* which models the completion of the loop—may fire. The duration of transition *End* is set to 0,since it's firing corresponds to a logical action rather than a computation. The role of place $L(P_i, x)$ is to enforce the execution of single iteration at a time.

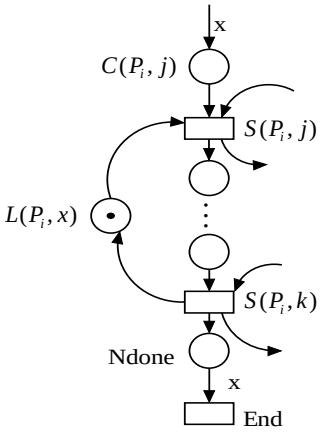


Fig. 2. The TTPN model of a known loop number involving several segments

Loops whose body contains an unknown loop number involving several segments are modeled as depicted in Fig.3.

In the Fig. 3. , all the loops are of the "while" type, the transition $S(P_i,1)$ represents that the judgment condition is true, the transition $S(P_i,2)$ represents that the judgment condition is false, the place *Ndone* is added to a token by each loop, then the transition $S(P_i,m)$ may fire, and deal with the data, the place $C(P_i,m)$ get a token. When the transition $S(P_i,2)$ represent that the condition is false, $C(P_i,2)$ will get a token, so the transition

$S(P_i,k+1)$ can fire, the "while" loop is completed. A parallel program may be divided into many logical processes, each process can be divided into segments, then according to the Fig. 1, Fig. 2, Fig. 3, we may obtain the timed Petri net model of the parallel program.

Regarding some data parallel questions processing, using the above simulation method, the TTPN model are defined according to assign the process number, these processes which is connected through the correspondence. Because these processes have the same operation but processing data is different, with the simulation method, it is advantageous to deal with the parallel program in the parallel machine, and enhance the processing performance.

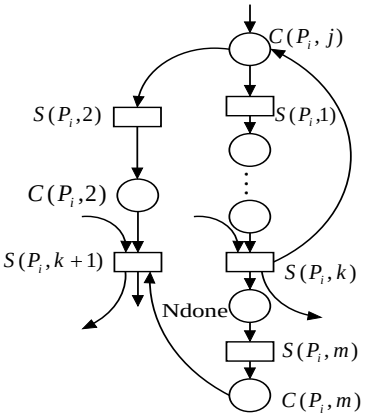


Fig. 3. The TTPN model of an unknown loop number involving several segments

4. MATRIX MULTIPLICATION ALGORITHM SIMULATION AND ITS REALIZATION IN DAWNNING 2000

Suppose A and B are both $n \times n$ matrices，compute $C = A \times B$ ，if we use m（m<n）processes， the pseudocode of process p_i about the matrix multiplication algorithm as follows：

```
size=n/m;
Send(B[i*size:(i+1)*size-1,0:n], (i+1)%m);
For(x=i*size;x<(i+1)*size;x++)
   For(y=0;y<n;y++)
      For(k=i*size;k<(i+1)*size;k++)
         C[x][y]=C[x][y]+A[x][k]*B[k][y];
For(j=1;j<m-1;j++)
   { receive(B[(i-j)%m*size:
   ((i-j)%m+1)*size,0:n],(i-1)%m);
if(j<m-1) Send(B[(i-j)%m*size: ((i-j)%m+1)*size,
   0:n],(i+1)%m);}
For(x=i*size;x<(i+1)*size;x++)
   For(y=0;y<n;y++)
    For(k=(i-j)%m*size;
        k<((i-j)%m+1)*size;k++)
          C[x][y]=C[x][y]+A[x][k]*B[k][y];
```

In the matrix multiplication (MM) algorithm, each process has the same operation but processing data is different, which completely conforms to the parallel question characteristic. The matrix multiplication algorithm has m processes, the process p_i (i=0,1, 2,…,m-1)is used to compute from the i*n/m-th row to (((i+1)*n/m)-1)-th row of the matrix *C*. Each process send from the i*n/m-th row to (((i+1)*n/m)-1)-th row of the matrix B to next process asynchronously, then enter the loop, until receive all message of the matrix B. According to the TTPN model of the Fig. 1. ,Fig. 2. and Fig. 3. , we can obtain the TTPN model of the process p_i in Fig. 4.

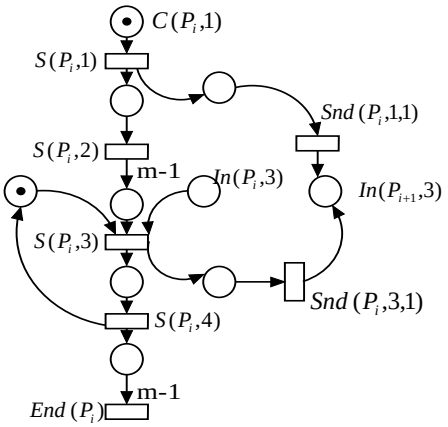


Fig. 4. The TTPN model of the process p_i

We also get the TTPN model of the matrix multiplication program including three processes (as shown in Fig.5). According to the model structure, we can partition the model into several processes, then mapping these processes to parallel machine. In the Dawning 2000 parallel machine, we use the simulation method to simulate two 400×400 matrices multiplication question, and carry out the comparison with the general inner product parallel method, the result is showed in Table 1. , and comparison chart in Fig. 6. We give out the speedup of the two methods Dawning 2000 in Fig. 7. The two chart show that the TTPN simulation method is better than the inner product parallel method in the MM algorithm.

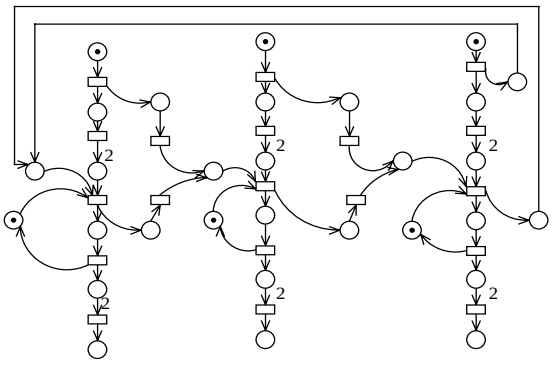


Fig. 5. The TTPN model of the matrix multiplication program including three processes

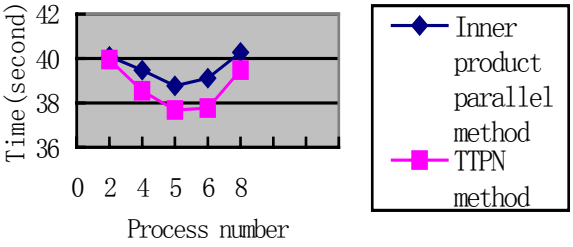


Fig. 6. The time graph of the two methods in Dawning 2000

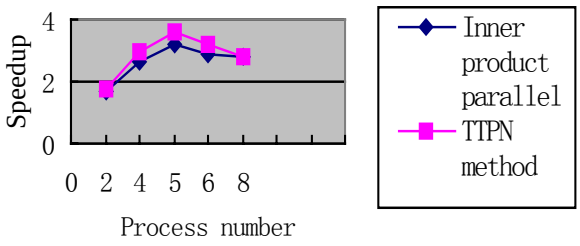


Fig. 7. The speedup of the two methods in Dawning 2000

Table 1. The execution time of the two methods in Dawning 2000(second)

Process number	2	4	5	6	8
Inner product parallel method	40.102317	39.476028	38.766327	39.114733	40.272354
TTPN method	39.952099	38.543052	37.674236	37.767108	39.472270

5. CONCLUSIONS

Regarding the data parallel question, it is essential to find the structure that has the same operation but processing data is different. This paper uses TTPN to analyze the data parallel question, and present the TTPN model of an unknown loop number involving several segments. We may find the structure that has the same operation but processing data is different by using the segment models. We give out the TTPN model of the matrix multiplication algorithm, and execute two 400×400 matrices multiplication question in the Dawning 2000 parallel machine, and carry on the comparison with the general inner product parallel method, the result shows he TTPN simulation method is better than the inner product parallel method in the MM algorithm.

REFERENCES

[1] T. Murata, “Petri nets - properties, analysis, and applications [J]”, Proc. IEEE, 1989,77(11): 541-580.
[2] J.C. Wang, “Timed Petri nets: Theory and Application”, Kluwer Academic Publishers, Boston, 1998.
[3] Monika Heiner, Dietmar Wikarski, “An Approach to Petri Net Based Integration of Qualitative and Quantitative Analysis of Parallel Systems”, BTU report I-09/1994.
[4] A.Ferscha, “A Petri Net Approach for Performance Oriented Parallel Program Design”, Journal of Parallel and Distributed Computing, July 1992,15(3): 188-206.
[5] G.Chiola, A.Ferscha, “Distributed Simulation of Timed Petri Nets: Exploiting the Net Structure to Obtain Efficiency”, Proc. of the 14th Int. Conf. on Application and Theory of Petri Nets, 1993.
[6] Cosimo Anglano, “Predicting Parallel Applications Performance on Non-dedicated Cluster Platforms”, Proc. of 12th ACM International Conference on supercomputing, July 1998: 172-179.
[7] J.W.Janneck, “Behavioral prediction of timed Petri nets with applications to distributed simulation”, Proceedings of High Performance Computing, Advanced Simulation Technologies Conference, 1998, 416-424.
[8] G.Chiola, A.Ferscha, “Exploiting Timed Petri Net Properties for Distributed Simulation Partitioning”, Proc of the 26th Howaii Int.Conf. On System Science, 1993, 194-203.

Grid-Based Parallel Data Mining

Jie Li
College of Mathematics and Computer Science
Fuzhou University, Fuzhou, Fujian, China
Email: mhxy-lijie@163.com

Xiufeng Jiang
College of Mathematics and Computer Science
Fuzhou University, Fuzhou, Fujian, China
Email: jxf1963@21cn.com

ABSTRACT

Knowledge discovery in datasets integrated into Grids is a challenging research task. These large datasets are being collected and accumulated across a wide variety of fields, at a dramatic pace. They are often heterogeneous and geographically distributed and globally used by large user communities. In this paper, we proposed the scenario that deploys a grid-based parallel data mining service in the Globus 4 Framework and hosts within Jakarta-Tomcat as a container in detail, using OGSA-DAI for access to Grid Data Source. The work presented in this paper reveals specific requirement for applying grid-based data mining in the service-oriented grid environment.

Keywords: grid service, service-oriented, Globus Toolkit 4, parallel data mining, OGSA-DAI.

1. IMPORTANT INFORMATION

Data mining, which aims to retrieve information automatically from large data sets, is one of the most important business intelligence technologies. Because of its high computational intensiveness and data intensiveness, data mining serves a good field of application for Grid technology.

The idea of data mining on the Grid is not new, but it has become a hot research topic only recently. The number of research efforts up to now is still quite limited. Many of the existing systems, such as NASA's Information Power Grid, Tera Grid and Discovery Net are either utilizing non-standard data mining techniques, or restricted to a special domain in the scientific realm [1].

Let's take a look at one of the prevalent issue in our real-life about the gird-based data mining: an advanced medical application addressing treatment of traumatic brain injury (TBI) victims. [2] TBIs typically result from accidents in which the head strikes an object. The trajectory of the TBI patient management includes the following main points: trauma event (e.g. injury at a car or sport accident), first aid, and transportation to hospital, acute hospital care, and home care. All these phases are associated with collecting data into databases, which are currently autonomously managed by individual hospitals, and are, therefore, geographically distributed. Moreover, they are heterogeneous and need high data security precautions. The aim is to use the Grid technology for building a virtual organization, in which the cooperation of the participating hospitals and institutions is supported by the integration of the above databases. Knowledge discovered in these databases can help to significantly improve the quality of the decisions taken by the health care specialists involved in treatment of the TBI patients. In most situations, a nearly real-time response to knowledge discovery queries is urgently needed. It is obvious that this kind of applications introduces new challenges to the Grid developers.

A grid enabled environment has the potential to solve this problem by providing the core processing capabilities with secure, reliable and scaleable high bandwidth access to the various distributed data sources and formats across various administrative domains. [3] GT4 is a set of software components that implement Web services mechanisms for building distributed systems. Web services technologies, and GT4 in particular, can be used to build both service-oriented applications and service-oriented infrastructure. Therefore, base on the idea of service-oriented, our infrastructure for parallel data mining on the gird afford as a series of grid service.

The rest of the paper is organized as follows. Section 2 describes a typical grid-based data access and integrated service for the large data from distributed and heterogeneous data sources. It is an indispensable preparative service to provide data for data mining service. We deploy it by using a free available middleware: OGSA-DAI and with essential security of Globus Toolkit 4.Section 3 introduces the process of parallel data mining and algorithm. Section 4 discusses the core components of the data mining service should include. Section 5 describes the main steps to deploy the service in Globus Toolkit 4 container. Finally, in section 6, we briefly conclude and outline the future work.

2. Data Service

2.1 Data scenario specific requirements

Grid-based data mining system should be able to cope with the heterogeneity of computers, operating systems, networks, but also that from the different sources of data and different data mining software. Therefore, we need a valid approach to deal with different possibilities of data distribution and access data from various kinds of data repositories. For the sake of described above, we need to design a mediator as a middleware service, which connects to the participating data sources, integrates them

This work was partially supported by Grid Portal Industry Technology Research and Development Project, under Grant Fujian Technical Committee and Data Interface of Scientific Computing in Grid Environment, under Grant Fujian Education Committee

logically into virtual data source (VDS), sends queries to them, and combines and delivers the results in a flexible way.

2.2 Implementation of the data service

In order to avoid building a new proprietary solution and reimplementing well solved aspects of Grid Data services, we have decided to extend the free available OGSA-DAI Grid Data Service (GDS)[4] reference implementation to provide a virtual data source (VDS). OGSA-DAI WSRF 2.1 is a WSRF-compliant version of OGSA-DAI and runs upon Globus Toolkit 4.0.2. It is not compatible with Globus Toolkit 3. The source release should still be compatible with Globus Toolkit 4.0.[5]

Offering the same metadata as a normal Grid data source, it can be used and integrated in existing applications quite easily to hide the distribution and heterogeneity of the participating data sources by reformulating requests against the logical schema of the VDS. All we need to do is just deploying the OGSA-DAI Data Service in the Globus 4 Framework and host within Jakarta-Tomcat as a container.

3. Data Mining Service

Data mining is the process of extracting information from large volumes of data. By its very nature, it is a computationally intensive exercise. Because data volumes are doubling annually, it would be difficult to achieve even small performance improvements simply by tweaking known data-mining algorithms. [6] Of course, sampling the input data can help, but the price is reducing accuracy, which is unacceptable for many tasks. Increasing the power of hardware doesn't offer much help; CPU clock frequencies and hard-drive data-transfer rates both have fixed upper boundaries, and no matter how much money you spend, you cannot overcome them. [7] Therefore, we must afford a parallel data minning service to deal with the large number of data and enable cooperative processing of single or multiple tasks by different computers.

3.1 The Process Of Parallel Data Mining

In the data minning service, two or more grid nodes simultaneously participate in the data-mining task. We divide these nodes into two types: one Central node and other Calculation nodes. The Central node is responsible for the scheduling and coordination of processes running on the Calculation nodes. There is only one Central node in a grid. Users prepare "jobs" and communicate with the Central node through grid client software. The client software lets users monitor and control job execution progress and examine and change grid configurations. User requests are sent to the Central node, which divides them into chunks that can be executed in parallel, sending these chunks for execution to all available Calculation nodes. Calculation nodes are responsible for the execution of tasks received from Central node. They report back to the Central node about execution progress and completion. Exchanging large chunks of data among the Central node, Calculation nodes, and client software requires shared data storage, such as a database server or shared file system.

3.2 Parallelization of Data-Mining Algorithm

In data mining, "prediction" is the task of producing unknown data values based on some previously observed data. Input data for prediction is divided into two parts—one containing training data and the second containing target data that to be predicted. Both parts have the same variables. Variables are divided into two groups: independent and dependent. Independent variable values are provided in both training and target parts, while dependent variables are known only in the training part. The goal is to predict dependent variables in target data. Here, we used a prediction method of data mining: nearest neighbor method. This method, by maintaining a dataset of known records, finds records (neighbors) similar to a target record and uses the neighbors for classification and prediction. The method is attractive because it needs only to do minimal preprocessing and predict each record independent of others, so modification is no necessary. With real-world problems, the number of records varies from thousands to hundred of millions. This method is particularly well suited for scalability. [8]

The algorithm is described as follows:

1.Divide the target data into the desired number of non-overlapping chunks. The number of these chunks must be great enough to take advantage of all the computational nodes available. On the other side, the chunks should be big enough so that the scheduling and preprocessing overheads are not significant compared to single data chunk processing time.

2.Copy training data to each Calculation node.

3.Process training data on each node.

4.Assign a chunk of target data to each Calculation node and perform predictions against the assigned chunk.

Feature selection algorithms are used to determine the minimal subset of variables upon which the variables of interest depend. Currently, we do not know any methods that can effectively parallelize feature selection for a single dependent variable. But often it is the case there are several dependent variables. In this case, each dependent variable can be processed on separate Calculation nodes, in parallel.

Validation is used to estimate how good the prediction is. To do this, training data is only thing to do. Some part of data is pretendedly unknown and will be predicted based on the rest of the records. Then the error of prediction can be estimated. The process usually tries several times to predict different parts of training data.

4. Data Mining Service Infrastructure

The data-mining infrastructure should include the following base components:

− GridMiner Service Factory (GMSF).

The GMSF is a specialized, persistant service for GridMiner-related services that is responsible for creating instances of transient GridMiner services.

− GridMiner Service Registry (GMSR).

The GMSR is a specialized, persistant service that is derived from the standard OGSA registry service.

− GridMiner DataMining Service (GMDMS).

The GridMiner DataMining Service is a central, transient GridService that provides an extensible set of data mining and data analysis algorithms and features.

− GridMiner Preprocessing Service (GMPPS).

The GridMiner Preprocessing Service encapsulates functionality that is applied before the main data-mining step. Such preprocessing activities include data cleaning, integration, handling missing data and statistical noise, aggregations, subset selection and many more.

− GridMiner Presentation Service (GMPRS).

The Presentation Service is a transient, thin component that receives a machine-readable form of a data-mining model as input and translates, renders or processes the model into different output formats.

Fig. 1. shows a sketch of the grid data mining service infrastructure that perform a parallel data mining algorithm.

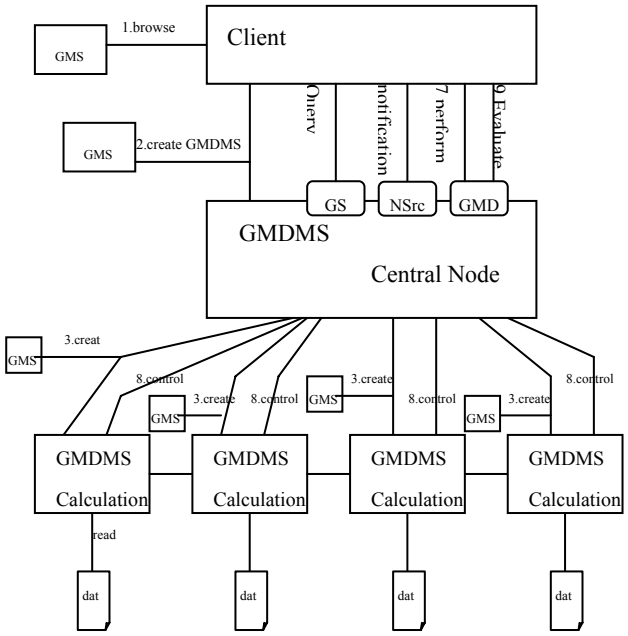


Fig. 1. A sketch of Parallel Data Mining Method

5. Deployment of Service With Globus Toolkit 4

The term "deploy" refers to the task of installing, and initiating the execution of a web service on a particular computer. Depending on the environment into which the service must be deployed, we need first to install and configure a GT4 container. The GT4 distribution typically provides convenient packages to facilitate various kinds of deploy tasks in different environments, so that the user is not forced to execute many installation and configuration steps to produce a system capable of running a specific service. The primary steps involved in developing and deploying Java services with Apache Axis are as follows:

1. Define the service's interface. In this paper, we define an interface file with Web Service Description Language (WSDL) describing the service's abstract interface.

2. Implement the service. We develop Java routines for the service implementation and use WSRF mechanisms for associated resource properties.

3. Define deployment parameters. A deployment file with Web Services Deployment Descriptor (WSDD) mechanism is used to describe various aspects of the service's configuration.

4. Compile all files described above and generate a GAR file. Compilation generates application specific interface routines that handle the demarshalling/marshalling of the Web service's arguments from/to SOAP messages.

5. Deploy the service.

As a parallel program is run across multiple distributed computers, execution management tools are necessary which are concerned with the initiation, monitoring, management, scheduling, and coordination of remote computations. GT4 supports the Grid Resource Allocation and Management (GRAM) interface as a basic mechanism for these purposes. The GT4 GRAM server is typically deployed in conjunction with Delegation and RFT servers to address data staging, delegation of proxy credentials, and computation monitoring and management in an integrated manner. Associated tools fall into three main classes. First, we have GRAM-enabled schedulers for clusters or other computers on a local area network. Second, we have systems that implement various parallel programming models in Grid environments by using GRAM to dispatch tasks to remote computers. Third, we have various "meta-schedulers" that map different tasks to different clusters.

6. Conclusions And Future Work

In this paper we have described our current research work, which focuses on the application and extension of the Grid technology to knowledge discovery in Grid databases, an important, but non-traditional Grid application domain. Because the young grid technologies has been developed just a few years, all the previous relative research done in the grid-based data mining field are connected with Globus Toolkit 3. And version 4 of the Globus Toolkit, GT4, released in April 2005, represents a significant advance relative to the GT3 implementation of Web services functionality in terms of the range of components provided, functionality, standards conformance, usability, and quality of documentation. So, there are something different in deploying our data mining service with GT4 from with GT3.In this paper, we proposed our scenarios to deploy the services in the Globus 4 Framework and host within Jakarta-Tomcat as a container, including data service and data mining service. In addition, we explant the data-mining algorithm into the grid environment and expand it support the parallelization.

The integration of two comprehensive research fields like data mining on the one hand and the younger and still evolving Grid technologies on the other hand is as ambitious as extensive. Furthermore, The design of a generic Grid-based data mining system should take full consideration of application scenarios both in scientific realm and business realm. On the way towards an open, service-based data mining system, we have already taken the important steps, but the succedent road is still full of brambles and more hardly.

7. REFERENCES

[1] T.C.Li, T.Bollinger, et al, Grid-based Data Mining in Real-life Business Scenario Germany, 2004.

[2] P. Brezany, A M. Tjoa, et al. GridMiner: An Infrastructure for Data Mining on Computatutional Grids, 2004.

[3] Data Mining Tools and Services for Grid Computing Environments (DataMiningGrid) Consortium. http://www.datamininggrid.org/

[4] OGSA-DAI. http://www.ogsadai.org.uk/

[5] G Toolkit. http://www.globus.org/

[6] A Depoutovitch, A Wainstein, Building Grid-Enabled Data-Mining Applications Tackling big Data-Mining jobs with parallelization, 2005.

[7] B. Gregory, S. Parthasarathy, et al, Towards Data Mining on Emerging Architectures, 2005.

[8] J. Han, M. Kamber, Data Mining Concepts and Techniques, 2003,183-187.

Research of Scheduling Algorithm Based on P2P Technology

Wei Zhao, Xihuang Zhang
School of Information Engineering, Southern Yangtze University
Wuxi, JiangSu, 214122,China
Email: shui_yun_zhaowei@163.com

ABSTRACT

Task scheduling is an integrated component of Grid computing among the key technology of the Grid. P2P_Grid model, a new Grid resource management model was brought up in this paper. In this model, we made some Super-Peers to join the Grid, they divided the Grid into corresponding sub-Grid. Each Super-Peer composes one local Grid system and answers for dealing with the local Grid events; these Super-Peers in different local Grid are equal and they change messages by P2P mode. A modified Grid scheduling algorithm was drawn out based on P2P_Grid model and the idea of traditional Min_min scheduling algorithm. At last, the new algorithm was compared with traditional Min_min algorithm by simulate experimentation. It shows that the new algorithm has a better quality of system load balancing and the utilization of system resource.

Keywords: Grid, Task Scheduling, Resource Scheduling, P2P, Super-Peer.

1. INTRODUCTION

Grid is a new network infrastructure that based on Internet, it is an important science and technology progress after Internet [1]. Grid technology has become one of the research hotspot in the computer fields. The idea of Grid comes from Power Grid, its target is that hope people to get high-powered computing capacity and share various heterogeneous resources as they using power-net in the daily life. But Grid has some characteristics such as distribute widely, numerous and complicated types, huge amount and dynamic etc. Because different Grid technology may be used in the different fields, it leads that resource distributing density is not symmetrical in the whole Gird. P2P and Grid are new style distributing computing models, their aims are similar, so it makes possible that setting up P2P_Grid model. P2P technology has advantage of load balancing and abundant information resources, thus we can use these advantages to make up for the fault of scatteration of Grid resources.

P2P technology is applied to the Grid and we adopt P2P protocol to deal with Grid Computing, namely join some Super-Peers into Grid and come into being a new Grid resource organization model-P2P_Grid model. It makes each center Super-Peer to form one local Grid System and answers for dealing with local tasks. Various local Grid systems are connected by P2P technology. In this model, for a group of clients, every Super-Peer is used as a Server, but it is equal between every two Super-Peer. The topology structure carries out the balance between centrality search and anonymous efficiency, it also keeps the robustness of load balancing and distributing search. In the Grid information server that based on Super-Peer model, each participant organization can configure one or more nodes and operate as Super-Peer. The nodes in each organization will change information of monitor and resource find, but for the nodes in the different organization will change information by P2P manner. At the same time P2P_Grid model has many characteristics such as independent of concentrate control, distributing, extension and can adapt resource state change etc.

We will give a new grid task scheduling algorithm in order to improve quality of the system load balancing and utilization of the system resource in this paper. The new algorithm is better logical and efficiency than conventional algorithms by simulation testing. There are six sections in this paper, the material content are as follows: the first section is Introduction; the second section is Review and contrast of existent grid task scheduling algorithms; the third section is P2P model and its inner management model; the forth section is Scheduling model; the fifth is the Simulation testing model and the testing result; the last section is Conclusion and Future work.

2. REVIEW PREVIOUS HEURISTIC SCHEDULING ALGORITHMS

In this section, we will review and contrast a set of heuristic scheduling algorithms in heterogeneous computing system. First, we mention meta-task concept: A meta-task is defined as a collecting of independent tasks with no data dependences. A meta-task is mapped onto machine statically; each machine executes a single task at a time. In general, we take into account a set of meta-tasks in the Grid environment. There are a large number of heuristic algorithms to be designed to schedule task to machines on heterogeneous computing system. In this section, we will list eleven basic heuristics scheduling algorithms as follows:

OLB: Opportunistic Load Balancing (OLB) assigns each task, in arbitrary order, to the next available machine [7, 8].

UDA: User-Directed Assignment (UDA) assigns each task, in arbitrary order, to the machine with the best-expected execution time for the task [7].

Fast Greedy: Fast Greedy assigns each task, in arbitrary order, to the machine with the minimum completion time for that task [7].

Min-min: In Min-min, the minimum completion time for each task is computed respect to all machines. The task with the overall minimum completion time is selected and assigned to the corresponding machine. The newly mapped task is removed, and the process repeats until all tasks are mapped [7, 10].

Max-min: The Max-min heuristic is very similar to the Min-min algorithm. The set of minimum completion times is calculated for every task. The task with overall maximum completion time from the set is selected and assigned to the corresponding machine [7, 10].

Greedy: The Greedy heuristic is literally a combination of the Min-min and Max-min heuristics by using the better solution [7].

GA: The Genetic algorithm (GA) is used for searching large solution space [6].
SA: Simulated Annealing (SA) is an iterative technique that considers only one possible solution for each meta-task at a time. SA uses a procedure that probabilistically allows solution to be accepted to attempt to obtain a better search of the solution space based on a system temperature [4].
GSA: The Genetic Simulated Annealing (GSA) heuristic is a combination of the GA and SA techniques [2].
Tabu: Tabu search is a solution space search that keeps track of the regions of the solution space, which have already been searched so as not to repeat a search near these areas [5].
A*: A* is a tree search beginning at a root node that is usually a null solution. As the tree grows, intermediate nodes represent partial solutions and leaf nodes represent final solutions. Each node has a cost function, and the node with the minimum cost function is replaced by its children. Any time a node is added, the tree is pruned by deleting the node with the largest cost function. This process continues until a complete mapping (a leaf node) is reached [3].

The testing results show that OLB, UDA, Max_min, SA, GSA, and Tabu do not produce good schedules in general. Min_min, GA, and A* are able to deliver good performance. The difference between the completions times of the schedules (*makespans*) generated by these three algorithms is within 10%. GA is consistently better than Min_min by a few percents, since it is seeding the population with a Min_min chromosome. A*, on the other hand, produces better or worse schedules than Min_min and GA in different situations. Among the three algorithms, Min_min is the fastest algorithm, GA is much slower, and A* is very slow. For 512 tasks and 16 machines, the running time of Min_min is about 1 second, GA 30 seconds, and A* 1200 seconds.

In fact, task scheduling of Grid is divided two processes, resource scheduling and task scheduling. The conventional task scheduling algorithms always study each scheduling. As a result, it must be led to ignore some existing parameters actually. For example, the time of scheduling resource is ignored in the task scheduling algorithm, similarly the attribute of scheduled task is ignored in the resource scheduling, for instance, QOS request /supply that scheduled tasks need.

The task scheduling algorithm that given in this paper, we take into account not only the attributes of tasks themselves but also the delay that resource scheduling is brought. So new algorithm will embed resource scheduling into task scheduling, and produce the new grid task scheduling algorithm.

3. P2P_Grid MODEL AND ITS INNER MANAGEMENT MODEL

3.1 Complementarity between P2P and Grid

Both P2P and Grid are distributing computing model, their success offers a good research and development example for distributing system [11]. So, P2P and Grid has become a research focus of the computer field now. In the development of P2P and Grid, Grid is the final aim, it gathers many distributing system technologies and tries hard to realize visit all resources that can share in the network now, but P2P is designed for making all computing communication entities to realize equal communication in the network now. So P2P computing overcome the fault that request of concentration computing need strong function network computers and expensive bandwidth cost, as a result it can increase system efficiency.

P2P system and Grid Computing are subset of distributing computing, as new-style distributing system, Grid and P2P model have many similar characteristics, thus, we can take advantage of the similar spots between P2P and Grid network, and we adopt P2P protocol and model to deal with Grid Computing. In a server example, a Super-Peer can call a Grid server by change one appointed information orders, it also may be call another Grid server by another Super-Peer, the Grid server is issued by another Super-Peer by one associated Grid server interface.

3.2. Super-Peer Inner Resource Model

In general, Super-Peer is constituted the region of resources concentrating in order to reduce communications delay when resources are scheduled. Fig.1 shows a Super-Peer local resource management model.

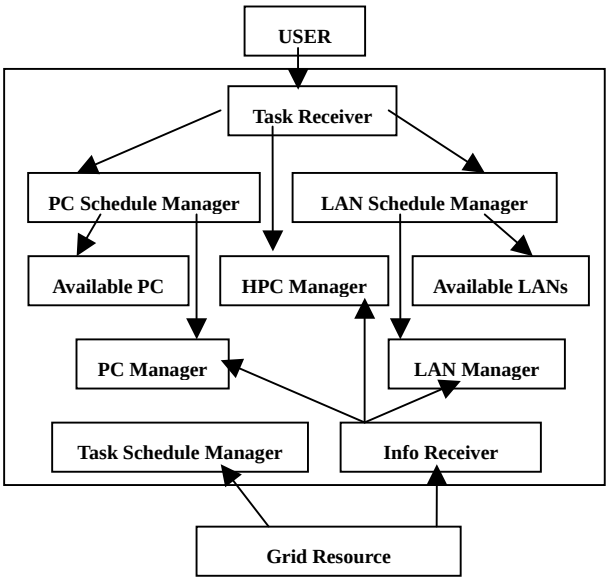


Fig. 1. Super-Peer local resource management model

In Fig.1, Grid Resource submits resource attribute information to request to register resources by Information Receiver. Information Receiver classes receiving resources and submits them to PC Manager, HPC（High Performance Computer）Manager or LAN Manager according to different sort. User submits task to Task Receiver, Task Receiver responds the task resource request when Task Receiver receives the task, and it submits lookup resource request to Resource Manager. Resource Manager searches those valid logical resources that registered by a certain searching algorithm, and registers these valid resources attribute information into the valid Resource Manager（resource ） according to a certain strategy and level order. At the same time, it must select the best resource or resource set assign to the submitted task.

Resource Receiver must control the thing of load in the system; it will transfer the task to other idle Super-Peer as soon as it discovers that loading of its local system is not balancing, so the new algorithm can overcome the fault of loading unbalance.

4. SCHEDULING MODEL

The task scheduling of Grid is divided two processes actually, resource scheduling and task scheduling. The conventional task scheduling algorithms always study each scheduling. As a result, it must be led to ignore some parameters of existing actually. For example, the time of scheduling resource is ignored in the task scheduling algorithm, similarly the attribute of scheduled task is ignored in the resource scheduling, for instance, QOS request /supply that scheduled tasks need.

The task scheduling algorithm that given in this paper is based on P2P_Grid model, thus, we take into account not only the attributes of tasks themselves but also the delay that resource scheduling is brought. Resource scheduling plays an important role among the whole task scheduling, a good resource-scheduling algorithm can increase load balancing and reduce the implement time of task scheduling. From the discussing of P2P_Grid inner management model, we can know that in this model, Super-Peer schedules resources among a local range, so it will avoid search resources from whole Grid range and make resource scheduling time to reduce, as a result, the scheduling time of all tasks will be reduced. Secondly, because P2P technology has advantage of load balancing, it will increase resources utilization of Grid system.

4.1 New algorithm idea

In our study, we embed the QOS request/supply into the grid-scheduling algorithm to make a better match among different level of QOS request/supply. In general, the practical request/supply of QOS is related to resource that tasks apply, for example, QOS for a network may mean the desirable bandwidth for the application; QOS for CPU may mean the requested speed, like FLOPS, or the utilization of the underlying CPU. In our study, a one dimension QOS is considered. Now, QOS request/supply is considered few in the most of existed scheduling algorithms. Consequently, while a task with no QOS request can be executed on both high QOS and low QOS resources, a task that requests a high QOS service can only be executed on a resource providing high quality of service. Thus, it is possible for low QOS tasks to occupy high QOS resources while high QOS tasks wait, as low QOS resources remain idle. As a result, it leads to reduce the system resource utilization.

We divide the scheduling takes into two set, high QOS request（*Task_H*） and low QOS request（*Task_L*）. Because new algorithm combines resource scheduling and task scheduling to deal with the grid task scheduling together, we must consider a significant parameter, that is resource scheduling time, let *ST* be the resource scheduling time. First we assume several parameters as follows: Let *ET* be expectation execution time of task, let *STAR*T be start execution time of task, so expectation completion time *CT* is defined as *CT=START+ET*, and *START=A+ST*; Let *A* be arrived time of task. We expect that *ST* should be enough short, thus the whole tasks scheduling time would be reduced. To get this aim, we introduce P2P technology into the new algorithm, each Super-Peer makes up of a local Grid system. When users submit tasks, they consider not only their task QOS request but also the Super-Peer that the submitted task belongs to. Users always submit the tasks to the Super-Peer, which they belong to. Super-Peer would deal with task scheduling after task is submitted. It can be divided three scenarios when Super-Peer schedules its own task, listed as follows:

1）If there are not other tasks that being executed or waiting for scheduling in the task queue memorizer of the Task Manager, when Task Manager receives the first this kind of task that user submits, it can search satisfaction demand resources in Resource Manager. Then, Resource Manager search whether existing valid resource queue in the corresponding resource group according to receiving resource. If resource queue is empty, search sub-tree of directory tree of resource group or leaf node, return collected information to Task Manager and Resource Controller. Resource Receiver divides the resources to different group based on the range of IP addresses and compounds them again after it receives information. These compounded resources must be saved at the same time. Resource Controller can search the best new state of these resources in the database and return it to Task Manager after Resource Controller receives the information about received resources. Task Manager makes resource scheduling last disposing based on receiving resource information twice. At last, Task Manager selects a group of the best quality resources to assign this task based on compounding and optimizing principle.

2）If there is other task or this kind of task to being executed in the task queue memorize that user submitted, when task queue is empty in the Task Manager and resource queue of resource group is not empty in the Resource Manager, because Resource Manager has saved a set of better QOS resources that has been completed this task when it receives this kind of task firstly, namely, there is resource queue that has been scheduled and is waiting for being assigned, thus, it is not necessary to submit searching resource request to Resource Manager. We can select those not be assigned resource group straight in resource queue of having been scheduled and return the information to Resource Controller. Resource Controller visits resource based on IP and locates resource at once, then, it must return to the new state of resource. Resource Manager can select a group of resource assign this task based on a certain compounding and optimizing principle after it analyzes the received information.

3）If there are other tasks to be waiting for resources and all this kinds of resources have been occupied in the task queue memorizer that user submitted, but this task must be completed in a certain time, thus Resource Manager can make this task to join the waiting queue, or submits it to the other Super-Peer, Super-Peer deals with it according to （1）or （2）that mentioned above after it receives this task, finally, return the result to user.

When user submits task to Super-Peer, they may meet such things that Super-Peer is not satisfied with QOS request/supply enough of user because local system loading is too much. The Super-Peer needs to transfer some tasks to other Super-Peer at once, in this case, we need to search whether exist Super-Peer of satisfaction demand and transfer tasks to it.

4.2 A new grid task scheduling algorithm

Every waiting for scheduling task has two attribute values: ①the Super-Peer that task belongs to; ②the type of QOS that task belongs to;

In the new algorithm, we still use the idea of Min_min to select waiting for scheduled tasks for the same kind of tasks that belong to the same Super-Peer. As a result, it ensures that expectation execution time of task *ET* is the shortest.

For the sake of simpleness, we divided all tasks into two sets (*Task_H* and *Task_L*) according to tasks' bandwidth request.

Algorithm procedure:

1）divide all submitted tasks into two sections according to

tasks' QOS request/supply and they belong to Super-Peer, place different set *Task_H* and *Task*_L;

2）for each task *Ti* in *Task_H* set, it is submitted the Super-Peer that it belongs to, Super-Peer deals with this task according to the 3 situations that mentioned above;

```
FOR   I=1 TO m   // m is a number of tasks in Task_H set
    {
IF(task_queue= =null AND resource_queue= =null)
                    // scheduling this kind of task firstly
{
Super-Peer searches resources that the task request;
Return the new state of resources;
Scheduling this task;
}
IF(task_queue==null AND resource_queue!=null)
//there is other task or this kind of task to being executed
{
 Super-Peer makes this task to join in the waiting queue;
Assign the resources to this task and waiting for scheduling;
Return the new state of resources;
}
IF(task_queue!=null AND resource_queue= =null)
Super-Peer makes this task to join in the waiting queue or
transfer it to other Super-Peer according to the urgent
situation of task;
}
```

（3）for each task Tj in Task_L set, the Super-Peer still works according to the above procedure.

5. EXPERIMENT RESULTS

We used SimGrid simulator [9] to make simulation testing for the new grid task scheduling algorithm by us. We put Super-Peer into SimGrid simulator and composed a local Grid system. For task QOS request, we considered bandwidth request only. QOS requests have a big effect on the performance of task scheduling. Based on actual applications, three different scenarios were used in our simulation testing, we set high QOS request ratio to 20%, 50%, 80% in all tasks. For each scenario and each heuristic we create 150 tasks 150 times independently and get the average makespan of the 150 times. Fig.2 and Fig.3 show the comparison between new algorithm and Min_min in the average makespan and system resource utilization.

As shown in Fig.2, we can get such conclusion as follows: where the tasks that require high QOS and the tasks that require low QOS are evenly distributed, the performance gain reaches as high as 15.74%. While where the tasks that require high QOS are in lower density (20%) or in higher density (80%), the performance gain are lower than evenly density(50%).As shown in Fig.3, we find that

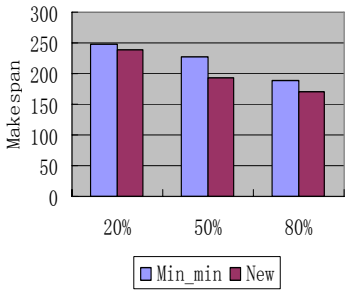


Fig. 2. Makespan for three scenarios for two algorithms

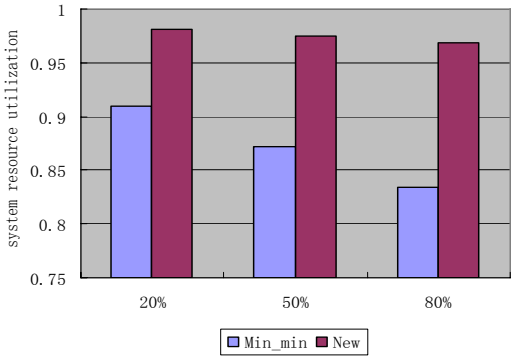


Fig. 3. system resource utilization for three scenarios for two algorithms

for the three scenarios, the system resource utilization as much high as Min_min. So, new grid task scheduling algorithm shows better quality than Min_min in two aspects.

6. CONCLUSION AND FUTRUE WORK

We first introduced several conventional task scheduling algorithms in heterogeneous computing system and compared them, discussed the fault of conventional algorithms deeply. We produced a new algorithm based on it. In our new algorithm, we embed P2P technology into Grid, compose local Grid system and take advantage of Super-Peer to complete local task scheduling. Comparing with basic heuristic Min_min algorithm, there have more improving in the quality of loading balancing and system resource utilization.

This study is a first attempt to embed QOS and P2P into the Grid task scheduling. There are more study spaces in this field, for example, embedding multi-dimensional QOS into task scheduling is still a topic of research; for Super-Peer, we hope find "the best fittest node "when the task is transferred other Super-Peer. So, it must produce searching strategy to find "the best fittest node" of Super-Peer.

7. REFERENCES

[1] Z. H. Du, Y. Chen, et al, Grid Computing[M] BeiJing Tsinghua University Publish,2002 .

[2] H. Chen, N. S. Flann, e al, "Parallel genetic simulated annealing: a massively parallel SIMD approach",IEEE Transactions on Parallel and Distributed Computing , 1998 ,9(2):126-136.

[3] K. Chow, B. Liu, "On mapping signal processing algorithms to a heterogeneous multiprocessor system", In ICASSP 91, 1991, 1585-1588.

[4] M. Coli, P. Palazzari, "Real time pipelined system design through simulated annealing", Journal of Systems Architecture, 1996, 42(6-7): 465-475.

[5] I. D. Falco, R. D. Balio, et al, "Improving search by incorporating evolution principles in parallel tabu search", In IEEE Conference on Evolutionary Computation, 1994, 823-828.

[6] L. Wang, H. J. Siegel, et al, "Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach", Journal of Parallel and Distributed Computing, 1997, 47(1):1–15.

[7] R. Armstrong, D. Hensgen, et al, "The relative

performance of various mapping algorithms is independent of sizable variances in run-time predictions", In 7th IEEE Heterogeneous Computing Workshop (HCW '98), 1998, 79-87.

[8] X. H. Sun, M. Wu, GHS: A performance prediction and task scheduling system for Grid computing, In Proc. of 2003 IEEE International Parallel and Distributed Processing Symposium.

[9] H. Casanova, G. Obertelli, et al, "The AppLeS parameter sweep template: User-level middleware for the Grid", In Proc. the Super Computing Conference (SC'2000), Dallas, Texas, Nov. 2000.

[10] R. Buyya, M. Murshed, et al, "A deadline and budget constrained cost-time optimization algorithm for scheduling task farming applications on global Grids", In 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02), Las Vegas, Nevada, USA, June 2002.

[11] B. Q. Zeng, Z. G. Chen, Research of Synergy between P2P and Grid[J],Computer Engineering and Applications, 1002-8331-(2005)11-0132-03.

Resource Management and Scheduling Model in Grid Computing Based on an Advanced Genetic Algorithm *

Hao Tian
School of Computer Science and Technology, Hubei University of Economics
Wuan, Hubei 430205, China
Email: haotian@mail.whut.edu.cn

Lijun Duan
Department of Computer and Information Engineering, Hubei Institute of Economics Management
Wuan, Hubei 430079, China
Email: duan_lj@sina.com

ABSTRACT

Grid computing is one of the research hotspots in high performance computing area. Efficient scheduling of complex applications in a grid environment reveals several challenges due to its high heterogeneity, dynamic behavior, and space shared utilization. We studied the characteristics of gird tasks, analyzed several main grid resource management models, and built a common model of grid tasks, put forward a hierarchy grid resource management and scheduling model, expounded the ideas of designing the model and described the details of it. Moreover, we proposed an advanced genetic algorithm (GA) in its scheduling strategy, particularized the principle and the function of this algorithm as well as giving the concrete plan to put every step of the scheduling strategy into practice. Finally, the algorithm was simulated with the aid of SimGrid toolkit. The results indicate that the advanced genetic algorithm could more efficiently realize global scheduling and could be an effective measurement for grid scheduling.

Keywords: Resource Management, Scheduling, Strategy, Model, GA.

1. INTRODUCTION

Grid computing is an important information technology that has growing up globally during the recent years. Its object is to build a universal and mass computing process virtual system consisted by several distributed resources including computation hosts, network bandwidth and data centers [1]. Grid technology has broad prospect of application in many fields, such as commerce, transportation, meteorology and education. There are great needs for high performance grid in many departments and enterprises which referring to scientific research, development and education.

Resource management is one of the key technologies in grid computing. It couples grid resources logically as a single integrated resource for users. Users can communicate with the grid system directly without considering the complexity of grid resources and grid architecture. Generally speaking, grid resource management system has three kinds of basic services: resource distribution, resource detection and resource scheduling.

Scheduling is an important part of grid computing. The efficiency and acceptability of resource management mainly depend on its scheduling strategy. Scheduling program allocates needed resources to the corresponding requests, including cooperation allocation through different systems. However, resource scheduling is becoming a complicated problem because of the dynamic and heterogeneous characteristics of grid system as well as the different needs for the resources of the applications applied in grid system.

This paper brings forward a hierarchy grid resource management scheduling model and adopts an advanced genetic algorithm (GA) in its scheduling strategy.

2. MAIN GRID RESOURCE MANAGEMENT SYSTEMS AND SCHEDULING METHODS

Architecture of the model of resource management mainly depends on the number of the resources that needed to be managed and the tasks that needed to be scheduled, and it also rests with whether the resources are in a single area or in several areas. Several grid resource management systems have been proposed in the last few years. There are three main kinds of them: centralized management model, distributed management model and hierarchy management model. Hierarchy management model is a mixed model (combination of centralized and distributed models), it synthesizes the characters of the two models, it not only can obey the local scheduling policy of resource owner, but also can manage whole system with the best scheduling method, so it's suitable for grid system. Agent technology is a hot topic in the distributed object-oriented systems. Agent can provide a useful abstraction on the grid environment. By their ability to adapt to the prevailing circumstance, agents will provide services that are very dynamic and robust, and it is suitable for a grid environment. Agents also can be used to extend existing computational infrastructures.

Grid scheduling strategy is the core of grid scheduling. There are so many special grid scheduling strategies such as the scheduling strategy Using Legacy Codes [2], the scheduling strategy based on LDS [3], the scheduling strategy Using Predicted Variance [4], Optimal Job Scheduling [5], PUNCH [6], XtremWeb [7] and so on. Most of them adopt a conventional strategy where a scheduling component decides which jobs are to be executed at which resource based on functions driven by system-centric parameters. The paper adopts the scheduling strategy based on an advanced GA.

3. GRID TASKS ANALYSIS

3.1. Characters and Model of Grid Tasks

* This work is Supported by the Educational Ministry of Hubei Province, China (Grand No.J200619006)

Different grid resource schedulers and different scheduling strategies have different views about grid task. Under the condition of taking complete time as optimization target, in the process of grid scheduling, we regard every user's request as a meta task, and partition a meta task into several independent tasks, viz. regard a meta task as a union of independent tasks and all tasks can be scheduled, but we don't exclude the dependence between every task. We regard a task as a combination of a data transfer subtask and a computing subtask, and we only consider the two subtasks using correlated resource to execute tasks. The model of meta task is showed in Fig. 1.

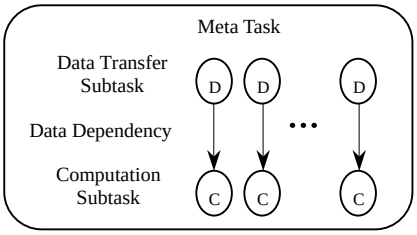


Fig. 1. Model of a meta task

In Fig. 1, data transfer subtask means the communication needed in computation subtask, viz. the part that needs expending store time and communication time. Computation subtask is the part of expending computing time. In fact, it's reasonable to partition task in current grid system and computing module. In the process of grid scheduling, a group of computers or mainframe computers are usually regarded as a computation resource (agent), because they are autonomous systems based on SSI (Single System Image), they have sound scheduling measures. After an application (meta task) is submitted to grid, grid scheduling program locates the data the meta task needs and assigns every task to its most suitable agent, and the agent executes the task after it gets the data. Changing the scheduling policy of a computer group or a mainframe is impossible and unnecessary [8], so we do not discuss whether there are subtasks in the two tasks.

3.2. Grid Task and Grid Resource

From what we have analyzed above, it is clear that every task relates to a data center and a computation agent (a group of grid resources) after scheduling. Data center first transfers the data that computation subtask needs to computation agent, and computation agent begins to execute the computation subtask as soon as the data transfer has finished, so the complete time of a task can be regarded as the sum of the complete time of its data transfer subtask and the complete time of its computation subtask.

4. MODEL OF RESOURCE MANAGEMENT AND SCHEDULING BASED ON AN ADVANCED GA

Fig. 2 shows the frame of our model, it includes four parts, viz. task partition module, scheduling decision module, information collection module and grid resource module.

4.1. Task partition module

Task partition module partitions a grid application (a meta task) into several independent tasks. There are many useful methods to do this. In this paper, it is assumed that any meta task can be partitioned into several independent tasks.

4.2. Scheduling decision module

Scheduling decision module uses the scheduling strategy based on the advanced GA to distribute every task to a corresponding group of computation agent and data center. It is the key component of the model.

4.3. Information collection module

Information collection module is composed of MDS [9] and NWS [9]. MDS is a grid information management system, it is used to collect and issue the state information of a system, we can gain much information from it: union of available resource agents, attribute of every agent, such as type of processor, speed of processor, amount of available processors etc. NWS is a distributed monitor system, it is specially designed to monitor resources in existence and network state, it can provide short-term network capability forecast, and it works on every agent so as to provide real time monitoring. We can get such data by using NWS: *availableCPU*, *currentCPU*, *bandwidthTcp*, *latencyTcp* and *connectTimeTcp* etc. Information collection module collects information about grid resources, and feedback it to the scheduling decision module, so as to provide evidence for scheduling strategy.

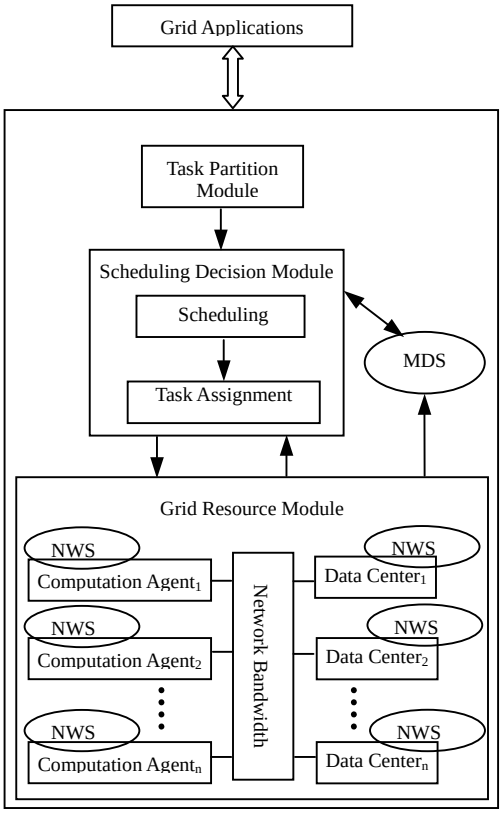


Fig. 2. Model of grid resource management and scheduling

4.4. Gird resource module

Gird resource module includes several heterogeneous computation agents, data centers and network bandwidth.

They are the hardware of a grid, and they can provide the resource that a meta task need.

5. SCHEDULING STRATEGY OF THE MODEL

We use the scheduling strategy based on an advanced GA. GA is a method of stochastic optimization and search, and it has the self-adapted search ability which has potential ability of learning. It expresses solution of problem as chromosome, before executing algorithm, it produces a group of chromosomes, viz. tentative solution, and puts the tentative solution into the environment of problem, then chooses the chromosomes can fit the environment from the group on the principle of survival of the fittest, produces a new generation group of chromosomes fit the environment better through crossover and mutation. After several generations anagenesis, there will be a new group of chromosomes fittest the environment, viz. the best solution of problem.

5.1. Description of the Problem

Based on previous discussion, it is assumed that a meta task is *T*, and it can be partitioned into *l* independent tasks, grid computation system is composed of n heterogeneous computation agents *C* ($C=\{C_0, C_1,\ldots C_{n-1}\}$) and *m* heterogeneous data centers *D* ($D=\{D_0, D_1,\ldots D_{m-1}\}$), every computation agent can get data it needs from any data center, then there are $m \times n$ communication lines (network bandwidths) in whole grid system, viz. $m \times n$ groups of grid resource. We express these groups as a $m \times n$ matrix *R*; express the transfer time from a data center to a computation agent as a $m \times n$ matrix *DT*; express the computing time of a task on a computation agent as a $m \times l$ matrix *CT*.

This shows, the essential of grid resource scheduling is distributing *l* independent tasks to $m \times n$ groups of grid resource in order to minimize complete time of a meta task and use grid resources sufficiently.

As the model shows, the value of matrix *DT* can be provided by NWS directly. The value of matrix *CT* can be calculated by Eq. (1).

$$CT(T_i, R(j,k)) = \frac{load(C_j)}{speed(C_j) \times processors(C_j) + 1}$$
$$(0 \le i \le l-1, 0 \le j \le n-1, 0 \le k \le m-1) \quad (1)$$

Where *speed* (C_j) is the average processor speed of agent *j*;
Processors (C_j) is the amount of processors of agent *j*;
Load (C_j) is the load assigned to agent *j*.

The three indexes can be gotten by NWS and MDS. Complete time of a task can be calculated by Eq. (2).

$$T_iT = DT(j,k) + CT(T_i, R(j,k))$$
$$(0 \le i \le l-1, 0 \le j \le n-1, 0 \le k \le m-1) \quad (2)$$

Complete time of a meta task can be calculated by Eq. (3).

$$MTT = \max\{T_iT \mid 0 \le i \le l-1\} \quad (3)$$

So the task of scheduling is to realize the best assignation of grid tasks on groups of grid resource to minimize *MTT*.

5.2. Initialization

The chromosomes coding technology in our strategy is subsection according to the amount of tasks, a section expresses a task, it forms from a task mark T_i and a resource group mark *R(j,k)*, it means that task T_i is executed by resource group *R(j,k)*. In order to create an initialization population, tasks are distributed to resource groups equally and stochastically.

5.3. Fitness Function

Choosing a proper fitness function can evaluate every iterative solution well. In this paper, the fitness function can be calculated by Eq. (4).

$$f = T_iT_c / (T_iT_m + 1) \quad (4)$$

T_iT_c is the complete time of task T_i with current scheduling strategy, and T_iT_m is the complete time of task T_i with the best scheduling strategy.

5.4. Selection and Anagenesis

We calculate the fitness function of task T_i of every generation, keep it if it fits the demand of convergence, then choose other tasks into choice union and realign their orders in chromosome.

Anagenesis is used to generate next generation chromosome. In the chosen sections of chromosome, it first tries different combinations of tasks within changeable limits, viz. tries to distribute the tasks to random grid resource groups once again, then calculates their complete time, and chooses the best scheduling combination, consequently finishes a time of anagenesis, the new chromosome will be the origination of next anagenesis, in this way, the global optimum solution will be found.

6. EVALUATION

We used the SimGrid [10] toolkit to evaluate our scheduling algorithm. The platform used for simulation is an example of grid model included in the SimGrid package.

We used this platform to simulate applications with different number of tasks (1000 and 10000 tasks) and quantity of computation per task (100,500, 1000 and 2000 MFlop/s) using deployments with 64 and 90 nodes (groups of grid resource). In our experiments we assumed that communication costs to send one task to an agent is fixed (0.001 Mbyte/s) and to receive the result is irrelevant.

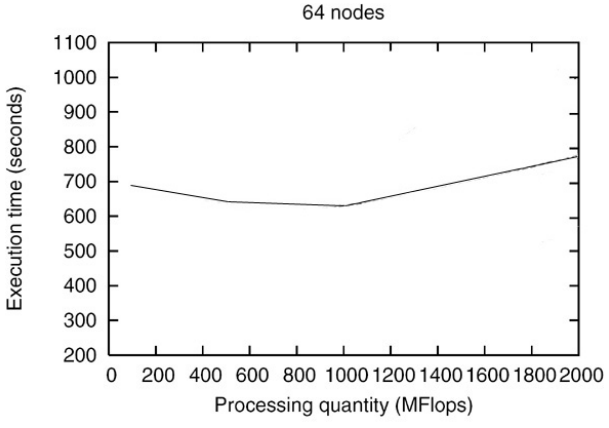


Fig. 3. Measurements scheduling 1000 tasks using 64 nodes

Fig. 3 illustrates the measurements obtained for an application containing 1000 tasks begin executed in 64 nodes of the platform, with computation amount per task

varying from 100 to 2000 MFlop/s. Using 64 nodes and computation quantity ranging from 100 to approximately 1000 MFlop/s, the model presented the best results.

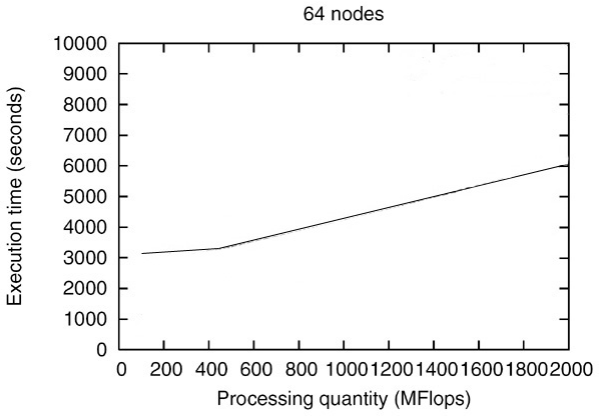


Fig. 4. Measurements scheduling 10000 tasks using 64 nodes

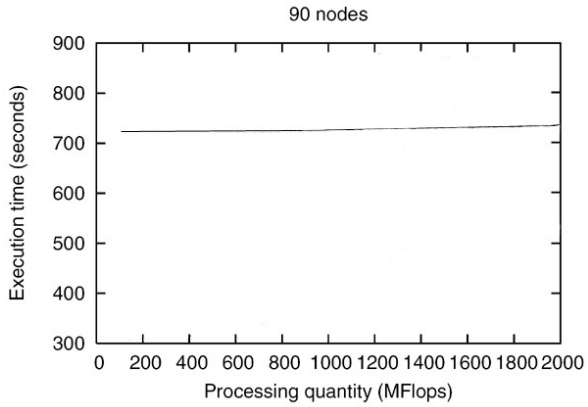


Fig. 5. Measurements scheduling 1000 tasks using 90 nodes

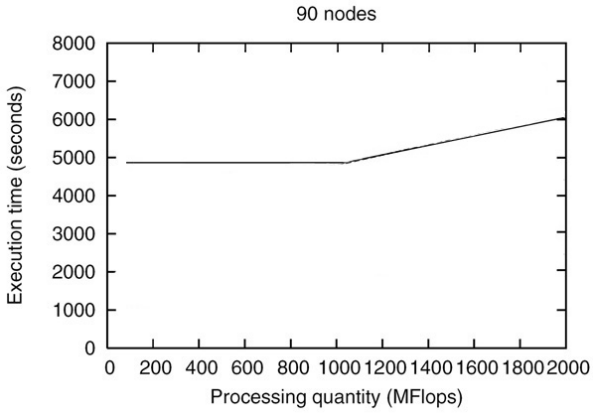


Fig. 6. Measurements scheduling 10000 tasks using 90 nodes

In Fig. 4 the same measurements are presented for an application with 10000 tasks. Using tasks with no more than 400 MFlop/s the model presented the best results. It behaved better from 200 to approximately 500 MFlop/s than after 500 MFlop/s.

Fig. 5 illustrates the measurements obtained for an application containing 1000 tasks begin executed in 90 nodes of the platform, with computation amount per task varying from 100 to 2000 MFlop/s. In Figure 5 the line is smoother than others in figur3 and figur4. Using 90 nodes and computation quantity ranging from 100 to 2000 MFlop/s, the model behaved stable.

In Fig. 6 the measurements are presented for an application with 10000 tasks executed in 90 nodes. Using tasks with no more than 1100 MFlop/s the model presented the best results.

It is clear that the model performs better with a higher number of tasks and groups of grid resource.

7. CONCLUSION

On the base of current research, we take the hierarchy management model as prototype, build a mended grid resource management and scheduling model, propose an advanced GA in its scheduling strategy, and simulate it by using SimGrid toolkit. The result shows that the model has good expandability, can realize globally optimum scheduling, it is an efficient plan of grid resource management scheduling. Its main limitation currently lies in the modeling of the grid capacities to treat the grid tasks. What we need to do is to improve the data modeling and the possibility to extrapolate the model for values of the workload that could be less regular. Nevertheless we believe this novel approach is promising and already a very good alternative to be considered when a scheduling algorithm is needed for scheduling applications in grid computing.

8. REFERENCES

[1] I. Foster, C. Kesselman, The Grid2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Press, pp.20-25, 2003.

[2] P. Kacsuk, A. Goyeneche, et al, “High-Level Grid Application Environment to Use Legacy Codes as OGSA Grid Services”, in the Proc. of the 5th International Workshop on Grid Computing (GRID 2004), 8 November 2004, Pittsburgh, PA, USA, IEEE Computer Society, pp. 428-435, 2004.

[3] Fabrício A.B. da Silva, et al, “A Scheduling Algorithm for Running Bag-of-Tasks Data Mining Applications on the Grid”, in Parallel and Distributed Processing and Applications: Second International Symposium, ISPA 2004, Hong Kong, China, December 13-15, 2004, Lecture Notes in Computer Science, vol. 314, pp. 254-262, Sept. 2004.

[4] L. Yang, J.M. Schopf, et al, “Conservative Scheduling: Using Predicted Variance to Improve Scheduling Decisions in Dynamic Environments”, in The Proceedings of the ACM/IEEE Supercomputing 2003 Conference, p.31, November 2003.

[5] J. Soldatos, E. Vayias, et al, “Grid Donors Resources Utilization Analysis towards Optimal Job Scheduling”, in the Proc. of the DPSN '04, Workshop, held in the Scope of IFIP Networking, 2004, Athens, May 14th 2004.

[6] R. Buyya, D. Abramson, et al, “Nimrod/G: An

Architecture for a Resource Management and Scheduling System in a Global Computational Grid", International Conference on High Performance Computing in Asia-pacific Region (HPC Asia 2000), Beijing, China. IEEE Computer Society Press, USA, 2000.

[7] G. Fedak, C. Germain, et al, "XtremWeb: A Generic Global Computing System", Proceedings of the 1st IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), May 15-18, 2001, Brisbane, Australia, IEEE CS Press, USA, 2001.

[8] L. Zha, Z. Xu, et al, "Grid Task Scheduling Simulations Based on Simgrid", Computer Engineering & Application, vol. 14, pp. 90-92, 2003.

[9] Y. Du, Y. Chen, et al, Grid Computing, Tsinghua University Press, pp.86-90, 2002.

[10] H. Casanova, "Simgrid: A toolkit for the simulation of application scheduling", in the Proceedings of the IEEE Symposium on Cluster Computing and the Grid (CCGrid'01), 2001.

Exploiting Coarse Grain Parallelism with Parallel Recognition Compiler

Lin Han，Jianmin Pang，Rongcai Zhao，Ning Qi
National Digital Switching System Engineering &Technological R&D Center
Zhengzhou, Henan China
E-mail: stroller_hl@163.com

ABSTRACT

How to analyze serial codes and automatically find its' parallelism for generating parallel codes to execute on large scale machines is a problem concerned by parallel recognition compiler. For communications between nodes by using message mechanism on distributed memory machines, we need to exploit coarse grain parallelism of serial codes to avoid losing parallel incomings due to expensive communications between computation nodes. In this paper, we present the major compilation passes used in our parallel recognition compiler. This compiler takes a sequential program as input and automatically translates it into parallel code, which can be compiled by the native parallel compiler and run on the target machine.

Keywords: parallel recognition compiler, parallelization analyses, computation and data decomposition, code generation.

1. INTRODUCTION

When generating parallel code for coarse-grained architectures, many delicate trades-offs must be managed. One of the most important things is to minimize the overhead of communication and synchronization while the balance of the load evenly across all the processors is kept. At one extreme, when the entire program is run on one processor, the absolute minimal overhead is accomplished, since there is no parallelism, there is no overhead involved in interprocessor communication and synchronization. Of course, the load balance and corresponding parallel speedup are very poor. At the other extreme, the best load balance is obtained when a program is decomposed into the smallest possible parallel elements, with the elements distributed evenly among the idle processors. With very small parallel program elements, no processor is stuck working on one large element while other processors are idle, awaiting work. However, the synchronization and communication overhead is maximized in this case. This overhead almost always outweighs the benefits obtained by the perfect load balance. Somewhere between these two extremes lies the most effective parallel decomposition—a parallelizing compiler is faced with the challenge of locating that sweet spot [1].

2. COMPILER OVERVIEW

The parallel recognition compiler takes a sequential program as input and automatically translates it into parallel code for the target machine. We use SUIF compiler system [2] as our implementation platform. The compiler takes sequential C/Fortran77 programs as input. The source programs are first translated into the SUIF compiler's intermediate representation. All program analysis and optimization passes operate on the intermediate representation, then it is converted into a combination of C and OPENMP/MPI parallel code, which is compiled by the native compiler and run on the target machine. The design of a complete compiler framework that incorporates parallelism analysis is shown in Figure 1.

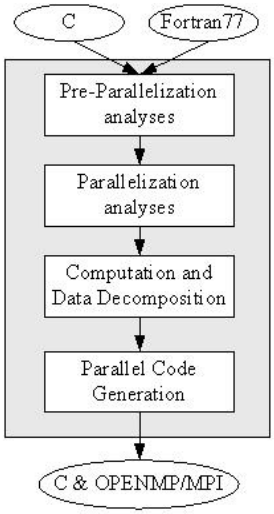


Figure 1: The design of parallel recognition compiler

The compiler first runs pre-parallelization analyses to gather information needed by the subsequent passes. The parallelization phase transforms the code to find the maximum degree of loop-level parallelism, and these loops are then passed on to the decomposition phase. The computation and data decompositions are used to generate parallel code for distributed address space machines. Finally, the compiler automatically generates SPMD (single program multiple data) codes with C and OPENMP/MPI statements.

3. PRE-PARALLELIZATION ANALYSES

The compiler runs pre-parallelization symbolic analyses to extract information necessary for subsequent parallelization and optimization passes. These analyses include scalar variable analyses such as constant propagation, induction variable identification, forward propagation and reduction recognition on scalar and array variables, and so on.

In this pass, the compiler also tries to transform the code so that each loop nest has as few array accesses as possible. Having fewer array accesses per loop nest reduces the possibility that the accesses will cause conflicting requirements on the computation decomposition for the loop nest. The loop fission transformation can be used to split a single loop into multiple loops that have the smallest number of possible statements. Each of the new loops has the same loop bounds as the original, but contains a subset of the statements [3]. After the decomposition analysis, loop fusion can be used to regroup compatible loop nests [4].

4. PARALLELIZATION ANALYSES

When attempting to enhance the parallelism extracted from a single loop, there are two general strategies that should be tried. If the loop is a sequential loop and carries dependence, finding some way to make it parallel is the obvious way to enhance parallelism. Any transformation that eliminates loop-carried dependences can be used to achieve this goal. Once a loop is parallel, increasing the granularity of the exposed parallelism is generally useful.

The parallelization analysis transforms the code using unimodular transformations to expose the maximum degree of loop-level parallelism, while minimizing the frequency of synchronization. It tries to generate the coarsest granularity of parallelism by placing the largest degree of parallelism in the outermost positions of the loop nest. Since no synchronization is needed between iterations of a parallel loop, pushing the parallel loops outermost reduces the frequency of synchronization.

The algorithm, developed by Wolf and Lam [5], to put the loop nests in a canonical form for consisting of a nest of the largest possible fully permutable loop nests are used in compiler. A loop nest is fully permutable if any arbitrary permutation of the loops within the nest is legal. A doall loop is simply a parallel loop and can execute in parallel with no synchronization. Loops that are distributed across processors but require explicit synchronization between iterations are called do across loops. We try to find parallelism of loops and mark the loops with doall and do across for the following decomposition analysis.

5. COMPUTATION AND DATA DECOMPOSITION

The decomposition analysis takes the loop nests in the canonical form of nests of fully permutable loop nests as input. It analyzes the array accesses within the loop nest to calculate the mappings of data and computation onto the processors of the target machine. For each loop nest and for each array accessed in each loop nest, the decomposition analysis outputs a system of linear inequalities that describes the processor mappings. The decomposition analysis only examines affine array accesses, and any non-affine accesses are ignored [6,7]. All affine array accesses within a loop nest are examined, regardless of control-flow within the loop nest.

A loop nest is said to be perfect if there is no intervening code between the headers of the loops, there is no intervening code between the tails of the loops, and the innermost loop does not enclose any loops. Perfect loop nests can be analyzed normally. Any non-perfectly nested accesses are treated as if they were perfectly nested, but with conditional guards in our compiler.

Now we focus on non-perfect loop nests and give the coarsest decomposition result. The following two examples show how we handle two kinds of non-perfect loop nests. Figure 2 shows how we handle a non-perfect loop nest, which has parallelism in the outer most loop indexed with k. After parallelization analysis, the compiler marks this serial code with doall and does across parallelism for every loop in this loop nest. Though the inner loop nests are parallel, we throw them away and only process the outer parallel loop for getting coarse grain decomposing result.

The decomposition result is given by our compiler in vector format as the following: computation decomposition is $(1)+(0)$, which means k-loop can fully parallel on a one-dimensional virtual processor space while the inner loops should be executed serially; data decomposition for array X,Y,Z are $(1\ \ 0\ \ 0)+(0)$, which means array X,Y,Z's first dimension can be distributed on a one-dimensional virtual processor space while the other two dimension should be serialized.

```
for(k = 0;k < N;k + +){                          //doall
    for(i = 0;i < N;i + +)                       //doall
        for(j = 1;j < N;j + +)                   //doall
            Y[k][i][N - j]+ = X[k][i][j];
    for(i=1;i < N;i + +)                         //doall
        for(j = 1;j < N;j + +)                   //doacross
            Z[k][i][j] = Z[k][i][j - 1] + Y[k][j][i - 1];
}
```

Figure2 : example of a non - perfect loop nest with parallelism in the outer most loop

Figure 3 shows how we handle a non-perfect loop nest, which has no parallelism in the outer most loops with index t. In this kind of serial code, the compiler turns to find decomposition of the inner loop nests for the outer most loops that cannot parallel.

```
for(t = 0;t < N;t + +)                           //do
{
    for(i = 0;i < N;i + +)                       //doall
        for(j = 0;j < N;j + +)                   //doall
            X[i][N - j]+ = Y[i][j];
    for(i = 1;i < N;i + +)                       //doall
        for(j = 1;j < N;j + +)                   //doacross
            Z[i][j] = Z[i][j - 1] + X[j][i - 1];
}
```

Figure3 : example of a non - perfect loop nest without parallelsm in the outer most loop

The decomposition result for the inner loop nests that our compiler gives in vector is as following, the first inner loop nest's computation decomposition is $(0\ \ -1)+(N)$, which means the first inner loop nest's second loop indexed by j can be partitioned on a one-dimensional virtual processor space; array X referenced in this loop nest is $(1\ \ 0)+(0)$, array Y is $(0\ \ -1)+(N)$, which means array X's first dimension can be distributed and array Y's second dimension can be distributed reversely in that one-dimensional processor space, where N means the offset. The second inner loop nest's decomposition is $(1\ \ 0)+(-1)$, and array X referenced in this loop nest is also $(1\ \ 0)+(0)$, array Z is $(1\ \ 0)+(-1)$. Here the compiler gives up finding the outer most loop's decomposition for it has no parallelism, and turns to decompose the inner loop nests. We also see array X referenced in both inner loop nests has an only decomposition vector, which means array reorganization communications for array X will not occur.

6. PARALLEL CODE GENERATION

Given a description of how the computation is to be partitioned across the processors, the compiler automatically produces an SPMD program to be run on each processor. The compiler generates the computation code and the necessary communication for each processor. Four kinds of communication codes should be generated, including data distribution communication, data alignment communication before computation, synchronization communication and data gathering communication.

We use a set of system named linear inequalities [8] to

represent computation decomposition, data decomposition and exact data flow information. The compiler solves the problems of communication code generation, local memory management, message aggregation and redundant data communication elimination by projecting polyhedral represented by sets of inequalities onto lower dimensional spaces.

7. EXPERIMENT RESULT

In this section we present experimental results for ADI integration. Our compiler pass took 2.64 seconds to generate the parallel code for ADI. Then we compiled it and ran on the SunWay cluster. This cluster consists of eight dual-processor 2.8 GHz Xeon processors, each with a 128KB first-level cache and a 16MB second-level cache, each processors node has 4GB memory, connected together by a 100M Fast Ethernet switch. Figure 4 shows the speedup of our experiment.

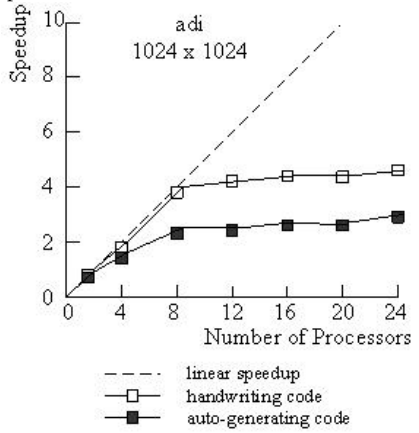


Figure 4: Speedup for ADI integration

Increasing the scalar of computation may improve the speedup in experiment. While up to now, the code generated may not be efficient and needs optimization to get more performance. There are several methods to optimize the communication such as message aggregation, message coalescing and overlapping communication with computation, and so on. These tasks will be carried out in our future work.

8. SUMMARY

This paper mainly describes the techniques we used in our parallel recognition compiler. The compiler takes sequential C/Fortran77 programs as input; the source programs are first translated into the SUIF compiler's intermediate representation. Then we use pre-parallelization and parallelization analyses to get loop-level parallelism. After giving the computation and data decomposition result presented by linear inequations, the compiler generates parallel codes with C and OPENMP/MPI statements. All algorithms presented in the paper have been used in our parallel recognition compiler.

9. REFERENCE

[1] R. Allen, K. Kennedy, Optimizing Compilers for Modern Architectures, A Dependence-Based Approach. ISBN 7-111-14122-9, 2001.

[2] R.P.Wilson, R.S.French, et al, SUIF: An infrastructure for research on parallelizing and optimizing compilers, ACM SIGPLAN Notices, 29(12): 31–37.

[3] D.Kuck, R.Kuhn, et al, Dependence graphs and compiler optimizations, Conference Record of the Eighth Annual ACM Symposium on the Principles of Programming Languages, Williamsburg, VA, January 1981.

[4] D.Callahan, A Global Approach to Detection of Parallelism, PhD thesis, Rice University, March 1987.

[5] M.E. Wolf, Improving Locality and Parallelism in Nested Loops, PhD thesis, Stanford University, August 1992.

[6] J.M.Anderson, M.S.Lam, Global optimizations for parallelism and locality on scalable parallel machines, Proceedings of the SIGPLAN '93 Conference on Programming Language Design and Implementation, Albuquerque, NM, June 1993, 112-125.

[7] J.M.Anderson, Automatic computation and data decomposition for multiprocessors, PhD thesis, Stanford University, March 1997.

[8] P.A. Saman, Parallelizing Compiler Techniques Based on Linear Inequalities, PhD thesis, Stanford University, June 1997.

Barrier Synchronization Optimizations for Compiler-Parallelized Programs

Ping Zhang , Qingbao Li , Rongcai Zhao , Jianmin Pang , Hongtu Ma
Department of Computer Science and Technology, University of Information Engineering
Zhengzhou, Henan 450002, CHINA
Email: lqb215@vip.371.net

ABSTRACT

This paper presents some compilation techniques for reducing barrier synchronization overhead in compiler-parallelized programs by using compile time knowledge of dependence and partition information. First, new parallel regions are reconstructed as large as possible by merging and expanding current parallel region recursively. Then by exploiting compile time computation partitions and communication analysis, cross-processor data dependence graph is constructed. Finally, a barrier optimization algorithm is developed. The algorithm bases on the graph theory and cross-processor data dependence graph is the basis of it. These optimizations have been implemented in a prototype parallelization compiler extending SUIF compiler. Evaluation using standard benchmark suites indicates that on average 37% of the barrier synchronization executed at run time is reduced by these optimization techniques.

Keywords: Synchronization, Parallel Region, Reconstruction, Cross-processor Dependence, Dependence Graph.

1. INTRODUCTION

Parallelizing compiler for shared-memory architecture generally realizes loop-level parallelization [1]. Tests indicate that load imbalance and barrier synchronization overhead are the key sources of inefficiency of the generated parallel programs [2]. While not much time is spent explicitly executing barriers, idle processor time in the form of sequential wait and load imbalance caused by barriers comprise a large percentage of total execution time, especially when barriers are in the inner loop. As the number of processors increases so does the relative overhead of barriers, making it a chief obstacle to scalability. It is clear from these measurements that reducing overhead caused by barrier synchronizations is important for achieving good performance.

We study barrier optimization in the context of a parallelizing compiler extending SUIF [3] to generate OpenMP [5] parallel program for shared memory machine. SUIF compiler is a parallelizing compiler infrastructure. It takes a parallel loop as a parallel region and the parallelized program executes in the fork-join model as shown in Fig. 1.

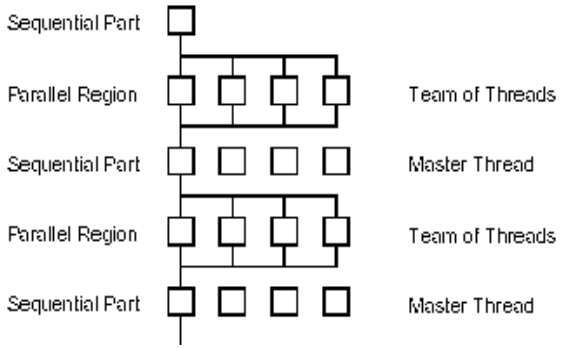


Fig. 1. Fork-join execute model

This model is flexible and can easily handle sequential portions of the computation; however, it imposes two synchronization events per parallel loop: a broadcast barrier is invoked before the parallel loop body to wake up available worker threads and provide workers with the address of the computation to be performed and parameters if needed. A barrier is then invoked after the loop body to ensure all worker threads have completed before the master can continue. This model introduces lots of barriers into target program. To reduce the barrier overhead, we realized some barrier optimization algorithms using compile time knowledge of dependence and partition information. These techniques include:

New parallel regions are reconstructed as large as possible by merging and expanding current parallel region repeatedly. Each new parallel region is the basis of barrier optimization.

Several program analysis measures are used to identify cross-processor dependence and build cross-processor graph.

In each parallel region, barrier optimization algorithm based on graph theory is applied to eliminate redundant barriers.

The remainder of the paper is structured as follows: Section 2 introduces our methods for reconstruct parallel regions. Section 3 discusses cross-processor data dependence analysis techniques. Section 4 presents barrier synchronization optimization algorithms based on cross-processor dependence graph. Section 5 is the results of evaluation. Section 6 introduces some related works. Section 7 concludes the paper.

2. PARALLEL REGION RECONSTRUCTION

Basic Process of Parallel Region Reconstruction

The aim of parallel reconstruct is to enlarge each parallel region to reduce the number of parallel regions and lay basis for barrier optimization. In fact, in the fork-join model, each parallel region has start up overhead such as broadcasts to the worker threads as well as overhead caused by end barrier. Thus reconstruction can reduce start up overhead at the same time.

The process of reconstruction includes two operations: mergence and expansion. Using the two operations, we construct new parallel regions as large as possible. Larger parallel regions may be constructed by following two rules:

Merge two adjacent parallel regions into a single parallel region.

Expand the parallel region to include the loop if the all statements in the loop body are in a single parallel region.

The rules are applied repeatedly until no further expansion of parallel regions is possible. During reconstructing, we keep barrier of the original parallel region, so the semantic of the program is not changed. After reconstruction, the number of parallel region is reduced but the number of barriers is not changed.

Notices in Reconstruction

In order to keep the correctness of the program, we need to take attention to the problem below during parallel region reconstruction.

Changes of Variable's Property: Reconstruction enlarges the range of parallel regions and may change the property of some data, such as turn shared variable to private variable. This may not affect sequential parts of the result program but may affect parallel parts because parallel executing need shared variables to communicate. In such case, the property must be modified. To solve this problem, we bring the definition of the variable whose property is changed out the current parallel region to recover its shared property.

Process of Serial Statements: The parallel regions reconstructed by the basic algorithm are conservative, no computations are replicated and processor utilization in the region is maximized because no sequential computation is enclosed. In theory, larger parallel regions may be created by allowing two additional types of statements to be included: replicated computations and guarded computations. Both replicated computations and guarded computations introduce additional overhead and result in lower processor utilization. As a result, these statements should be put in a parallel region only when necessary to enable merging, such as when they lie between two parallel regions or are the only statements in a loop body not in a parallel region.

Index Variables Recover: When a loop is expanded into a parallel region, the loop will be executed by all threads. If the index variable is shared, the result will be incorrect because threads affect each other when they access the same index variable. So such index variable should be privatized. To solve this problem, we define a new private variable to replace the shared one in the parallel region and assign the value of the shared variable to the new private variable. At the end of the parallel region the result is assigned back to the shared variable.

3. CROSS-PROCESSOR DEPENDENCE ANALYSIS

After parallel region reconstruction, each parallel region which could be looked as a small SPMD program [4] may contain multi parallel loops and some assign statements. The default barriers at the beginning and at the end of each parallel loop are still kept. Many of them are redundant. In order to eliminate redundant barriers, we need to know if there exist cross-processor dependence (the source and the sink are partition onto different processor) [14] between two statements. Based on the dependence and partition information given by previous passes, we use communication analysis to decide cross-processor dependences.

Communication analysis [7] is a set of techniques used to track the flow of data between processors. It was first developed in compilers for distributed-memory machines in order to identify where messages must be inserted into a program to access data on other processors. Here we incorporated ideas from distributed-memory computing to analysis cross-processor data dependence. If it can identify the producers and consumers of all data shared between two statements to be identical (i.e., the same processor), then there is no cross-processor dependences.

Our communication analysis algorithm based on liner inequality system. If two statements in a parallel region have dependence, we build communication set for each pair of array references causing the dependence.

***Definition1*:** If $(\vec{i}_1, \vec{p}_1, \vec{i}_2, \vec{p}_2, \vec{a}) \in (I \times P \times I \times P \times A)$ is an element of communication set M, if $\vec{p}_1 = C_1(\vec{i}_1)$, $\vec{p}_2 = C_2(\vec{i}_2)$, $\vec{a} = f_1(\vec{i}_1) = f_2(\vec{i}_2)$, at least one of f_1 and f_2 is write and $\vec{p}_1 \neq \vec{p}_2$. Where $\vec{i}_1$, $\vec{i}_2$ are iterations of two loop-nests, $f_1(\vec{i}_1)$ and $f_2(\vec{i}_2)$ are two array access functions, $c_1(\vec{i}_1), c_2(\vec{i}_2)$ are computation partitions of two regions. $\vec{a}$ is subscript of array element.

According to the definition of communication set, we can construct a symbol liner inequality system as shown in Fig. 2, where the last two inequalities are boundary of two-loop nest. We then determine whether the system of inequalities is consistent by scanning the system using Fourier-Motzkin elimination [13]. If a solution exists, cross-processor dependence exists, else not.

$$\begin{cases} \vec{p}_1 = C_1(\vec{i}_1) \\ \vec{p}_2 = C_2(\vec{i}_2) \\ f_1(\vec{i}_1) = f_2(\vec{i}_2) \\ p_1 \neq p_2 \\ b_1(\vec{i}_1) \geq 0 \\ b_2(\vec{i}_2) \geq 0 \end{cases}$$

Fig.2 Inequality system of communication set

4. BARRIER OPTIMIZATION

We study barrier optimization in each parallel region after reconstruction. The aim of our algorithm is to place as fewer barriers as possible in a parallel region. We consider globally how to place barriers in a parallel region. The algorithm is based on the fact that if there is a dependence between two statements then a barrier on the execute path form the source to sink of the dependence will ensure the correctness of the program and the location of the barrier is flexible, so in many case a barrier will satisfy several dependences at the same time.

Our algorithm base on the graph theory and cross-processor data dependence graph formulation is the basis of our algorithm.

Definition2: Cross-processor dependence graph CDG= (S, E) is a directed graph, where S are the vertices corresponding to statements, vertices in S can be divided into two types: control vertices and states vertices and E are the edges corresponding to cross-processor data dependences of a program. Such data dependences may be a flow (from a write reference to a read), output (from a write to a write) or anti (from a read to a write) dependences [13] whose source and sink are on different processors. In CDG, for clarity, we represent vertices of the CDG corresponding to statements by circles; vertices corresponding to controls structure such as *beginfor* and *endfor* by boxes and edges illustrate dependence relation of vertices.

We number all statements in a parallel region, including control structures, according to their lexicographic order. This can simplify our algorithm when we deal with a loop nest. We know that the statements whose numbers are between the number of *beginfor* and *endfor* construct the loop body. This gives a total order to vertices within basic blocks and loop nests and we can determine which vertex or statement precedes another simply by comparing their numeric values. A dependence whose source is statement *i*

and sink is statement *j* corresponds to an arc $i \rightarrow j$ in the CDG.

The overall barrier placement algorithm, shown in Fig. 3, is a bottom-up approach. This process is applied recursively until the entire parallel region is considered. The algorithm includes two main components: barrier optimization of loop and barrier optimization of basic block while loop optimization is based on basic block optimization.

```
BarrierOptimize(CodeSeg)
{
  for each statement CS in CodeSeg
  {
      if CS is the most inner loop
      {
          Loopbarrier(CS);
          consider CS as a statement;
      }
      Else  BarrierOptimize（forBody）;
  }
  Apply BlockBarrier()on all statements in CodeSeg;
}
```

Fig. 3 Barrier Optimization algorithm

Define 3*:* Let L = L1, L2, . . ., Lm be a loop nest of depth m, where *Lk* is defined as the loop at level *k* [1], whose graph contains a header denoted by *beginfor(k)* and an exit by *endfor(k)*.

Dependences can be divided into two types: loop independent dependence and loop-carried dependence. Dependences with a source and sink within the same iteration are defined as forward dependences. A loop-carried dependence has its source and its sink in different iterations of a loop nest. Furthermore, a loop-carried dependence at a particular loop level, with source *i* and sink *j*, is defined as either a *backward* dependence, if the sink is lexicographically before the source ($j < i$), or a wrap-around dependence, if the sink is lexicographically after ($j \geqslant i$).

In this paper we refer to barrier satisfying dependence in the sense that the insertion of such a barrier guarantees that the ordering of a program required by the dependence is satisfied and that we need no longer consider the satisfied dependences.

Preprocess

Form cross-processor graph, we can derive directly the facts:

Property 1: For all those forward dependences whose source is at vertex *i*. (See Fig. 4.), let $a_{i,1}$ be the sink of the dependence nearest to *i.* A barrier between statements *i* and $a_{i,1}$ will satisfy all the dependences whose sources is *i*.

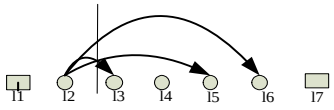


Fig.4 an example of property 1

Property 2: A backward dependence where *i* is the source of the dependence and *j* is the sink, may be satisfied by any barrier inserted between *i* and *endfor(k)* or between *beginfor(k)* and *j*. A wrap-around dependence at level k may be killed by any barrier inserted between *beginfor(k)* and *endfor(k).*

Property 3: If there are two dependences $i \rightarrow j$ and $u \rightarrow v$, where the source *i* is before the source *u* and the sink *j* is after the sink *v*, then $i \rightarrow j$ is covered. Barrier satisfying $u \rightarrow v$ will satisfy $i \rightarrow j$, so $i \rightarrow j$ can be removed.

Based on these properties, we firstly do dependence transform and dependence elimination to reduce complexity of barrier optimization.

- Based on Property 2, A backward dependence at loop level k, $i \rightarrow j$, is transformed to a forward dependence *beginfor(k)*$\rightarrow i$. A wrap-around dependence at loop level *k*, $i \rightarrow j$, is transformed to a forward dependence *beginfor(k)*$\rightarrow$ *endfor(k).*
- Based on Property 3 remove all covered dependence.
- For each vertex *i*, based on property 1, remove all the dependence except the dependence whose sink nearest to *i.*

Above measures reduce the number of dependences to be considered and reduce the complexity of the followed algorithm and reduce the space needed.

Barrier placement

After preprocess, only forward dependences are left in a loop. Thus the loop can be looked as a basic block. Our algorithm place the least number of barriers [6] in each loop and removes exterior dependences, which could be satisfied by these barriers placed in the loop. This reduces the total number of barriers.

Define 3: If both source and the sink of cross-processor dependence are in the same loop, we call the dependence interior dependence of the loop. If at least one of the sink of source of cross-processor dependence is not in the loop, we call the dependence exterior dependence of the loop.

After preprocess, each loop has the following property:

property 4 If t1 is the first sink after a vertex s, then placing a barrier just before t1 will kill all those dependences with sources between s and t1,see Fig. 5..

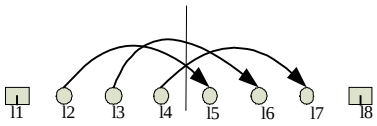


Fig.5 Example of property 4

Based on property 4, the idea of our algorithm to place barriers in a loop can be summarized as follows:

①Start at vertex *beginfor*, let *i*=1, *si=beginfor*;

②Find the first sink *ti* whose sources are at or after si. If *ti* exists then place a barrier just before *ti* and let *si+1=ti*, else goto ④

③Eliminate all dependence whose source before t_i and sink after t_i include exterior and interior dependence. Let i=i+1,goto ②

④Finish.

In step ③, exterior dependence is satisfied by barriers in inner loop. This reduces the total number of a parallel region.

For instance, in Fig. 6., there is five interior dependences in loop L: three interior dependences, l2→l3, l2→l5, l5→ l7,one backward dependence L6→L4 and one wrap-round dependence L3→L5.After preprocess, only two interior are left(Fig. 7.), using barrier placement algorithm, two barrier will satisfy these dependence: one is before L3 and one is before L7. At the same time, the first barrier satisfies the

exterior dependence l3→L6.

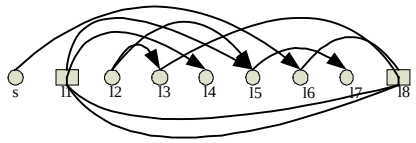


Fig.6 Cross-processor dependence graph of loop L

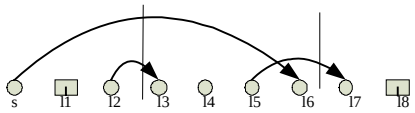


Fig.7 Cross-processor dependence graph after preprocess

5. EXPERIMENT RESULTS

We use ppopp benchmark suite of SUIF workgroup to evaluate our work. The set includes six programs: adi1k, erle64, lu1k, sor, swm256 and tomcatv. Table 1 shows the number of barriers executed in each program before optimization and after optimization. We see that on average about 37% of barriers executed can be eliminated by our algorithm.

Table 1 Experiment result of barrier optimization

Program	Execute barriers		
	Before optimization	After optimization	Elimination（%）
adi1k	42	16	47.6
ierle64	76	39	48.6
lu1k	2050	1027	49.9
sor	202	153	24.7
swm256	32	25	31.3
tomvatv	18	14	22
average			37.35

6. RELATED WORK

Synchronization is one of the key overheads in parallel computing and techniques to minimize its impact have been the area of research for a number of years. There are some techniques have been studied.

Program transformations to eliminate or decrease the cost of synchronization associated with loops have been proposed by several researchers. Loop fusion is used to avoid synchronization between loop nests while inverse transformation, loop distribution is used to move synchronization within a loop to a less expensive one between loops. Loop interchange [1] is used to place a parallelized loop outermost and loop alignment [13] is used to eliminate cross-processor dependences and hence synchronization.

There are other works concentrated on using less expensive forms of synchronization than the barrier. Data and event synchronization using post/wait statements have been extensively used to synchronize between loop iterations [9] and data elements [8] [10]. Runtime methods, such as fuzzy barriers, also have been used to reduce the cost of synchronization.

In [11], a graph-based barrier placement approach is proposed .It can place optimal barriers in basic block, perfect loop nest and a special case of imperfect loop nest but their algorithm is very complex for it need to deal with different structure separately. Our work is similar the idea of [12], but our approach is more simple and at most case can reach or exceed the performance of [11].

7. CONCLUSION

Reducing synchronization overhead in parallel programs is essential for scalable performance. In this paper, we presented compiler optimization techniques for reducing barrier synchronization overhead for compiler-parallelized program. Elements include parallel region reconstruction, cross-processor analysis and barrier placement. Our compiler optimization has been implemented in our parallelization compiler extending SUIF; we evaluate its performance using benchmark suites. Results show on average 37% of the barrier synchronization executed at run time is eliminated; some programs achieve dramatic reductions. Next we will study other optimization techniques to improve performance of parallelized program and it will become increasingly important as microprocessors out speed interconnection network and multiprocessor workstations become common.

8. REFERENCES

[1] R. Allen, K. Kennedy, Optimizing Compiler for Modern Architectures-a Dependence-Based Approach. Elsevier Science, USA, 2001

[2] C.Wen, Tseng, “Compiler Optimizations for Eliminating Barrier Synchronization”. The 5th ACM Symposium on Principles and Practice of Parallel Programming (PPOPP’95), Santa Barbara, CA, 1995.

[3] R. P. Wilson, R. S. French, et al, “SUIF: An infrastructure for Research on Parallelizing and Optimizing Compilers”. US: Computer Systems Laboratory Stanford University, 1994.

[4] G. Krawezik, F. Cappello, “Performance Comparison of MPI and three OpenMP Programming Styles on Shared Memory Multiprocessors”. SPAA'03, June 7.9, 2003, San Diego, California, USA, 2003.

[5] OpenMP Architecture Review Board. OpenMP C and C++ Application Program Interface.

[6] P. Zhang, R.C. Zhao, et al, “A Barrier Synchronization Optimization Algorithm Based on Cross-processor Dependence ” . Computer Engineer. 2005.09

[7] S. Amarasinghe, M. Lam, “Communication optimization and code generation for distributed memory machines”, Proceedings of theSIGPLAN’93 Conference on Programming Language Design and Implementation, San Francisco, CA, June 1992.

[8] P. Tang, P. Yew, et al, “Compiler Techniques for Data Synchronization in Nested Parallel Loops”, Proc. the 1990 ACM International Conference on Supercomputing, Amsterdam, the Netherlands, June 1990.

[9] D.K. Chen, Compiler Optimizations for Parallel Loops with Fine-Grained Synchronization, Ph.D Dissertation, University of Illinois at Urbana-Champaign, 1994.

[10] P. Tang, J.N Zigman, “Reducing Data Communication Overhead for Do across Loop Nests”, Proc.

International Conference on Supercomputing, Manchester, July 1994.
[11] E.A. Stohr, M.F.P. O' Boyle, "A Graph Based Approach to Barrier Synchronization Minimisation", Proc. International Conference on Supercomputing, July 1997.
[12] M. Gupta, E. Schonberg, "Static Analysis to Reduce Synchronization Costs in Data-Parallel Programs", Proc. Principles of Programming Languages, January 1996.
[13] H. Zima, "Supercompilers for Parallel and Vector Computers", ACM Press, 1991
[14] M.F.P. O' Boyle, L. Kervella, et al, "Synchronization Minimization in a SPMD Execution Model", J. of Parallel and Distributed computing 29, pp. 196-210, 1995.

GMM: a Grid-based MVC Model for Distributed Service

Chunhua Ju, Zhongyong Kong
Computer and Information Engineering College,ZheJiang GongShang University
HangZhou, ZheJiang Province, China
Email: juchunhua@hotmail.com, ayongs@21cn.com

ABSTRACT

Grid technology is becoming more and more attractive for coordinating large-scale heterogeneous resource sharing and problem solving. How to manage and maintain grid services which will be flooding on the web in the future is a crucial project. There is a highly available model which is based on the Globus Toolkit after analyzing the advantages and disadvantages of traditional technology. This model distributes each tier of the MVC model to distributed servers both physically and logically, enables different servers take different responsibilities. Each server has different kinds of mission and shares its services with others. There is a clear structure and explicit service and the maintenance will be easy in the entire grid system.

Keywords: Grid, MVC, Globus, Distributed, Service.

1. INTRODUCE

The concept of electronic commerce is put forward in 1994. Now electronic commerce is turning into a new stage----when there is a need, there is a pay, which is supported by the technology of grid.

Grid is becoming more and more attractive for coordinating large-scale heterogeneous resource sharing and problem solving. The user will be able to accomplish the mission in a powerful VOs[1,2](Virtual Organization) without costing much in the future.

Can imagine, the impact that grid technique brings is huge for the electronic commerce, although the grid technique places almost in the beginning stage. Using uniform services provided by the grid system is imperative under this situation. OGSA [3] requests us to write the service according to the norm plait, providing uniform interface. But various grid services, which are mutually applied, will make the logic of business mix. It will be hard to manage and maintain the grid services.

How to manage and maintain grid services, especially in the intranet, which will be flooding in the web in the future, is a momentous project. The following text describes several points about traditional technology on chapter2 and then the GMM (Grid-based MVC Model for Distributed Service) supported by Globus Toolkit on chapter3. The Globus Toolkit is an open source software toolkit used for building Grid systems. It has been widely adopted.

2. TRADITIONAL TECHNOLOGY AND PROBLEMS

Three-tier architecture

Foundation Item: Supported by the grand foundation of National Social Science Fund under Grant No 05BTJ019 and foundation of National Doctor Fund under Grant NO 20050353003.

The most popular style of web service in Internet now is three-tier architecture[4]which is a system enforcing a general separation among the following three parts

1) Client Tier or user interface.
2) Middle Tier or business logic.
3) Data Storage Tier.

It has resolved some limits of the two-tier architecture, but the weakness of the system is as follows: lack of transplant, bad compatibility of provider, narrow of adoption[5],and what more is the services can not be shared.

MVC Model

There are some basic concepts about MVC model before discussing the implementation and design of GMM.

Several problems may arise when applications contain a mixture of data access code, business logic code, and presentation code. Such applications are difficult to maintain, because interdependencies among all of the components cause strong ripple effects whenever a change is made anywhere, etc. The Model-View-Controller (MVC) design pattern solves these problems by decoupling data access, business logic, and data presentation and user interaction.

But MVC framework only achieves separating logically. The parts of model and controller are in the same server physically. That is not fit for the share of resource, and the following model distributes the Model tier, View tier, Controller tier to different servers. Each server is responsible for controlling disparate services. And all servers distributed on different places are controlled with grid tool.

3. GMM

After analyzing some traditional technology and models, there is a new model (GMM) based grid given. It uses grid technology to separate MVC model, not only logically but also physically, into different services. It makes the business resource easy manage and maintain.

Framework of GMM

The gird toolkit must be needed for the physical separateness of MVC model and the controlling of the distributed servers. The infrastructure of GMM is the Globus Toolkit [6,7]. The Globus Toolkit is an open source software toolkit used for building Grid systems and applications. It is being developed by the Globus Alliance and many others all over the world. Globus Toolkit is an implementation of the Open Grid Services Infrastructure (OGSI [8]) Version 1.0.

All business resource is divided into three kinds of service in GMM. The first is data service showing as Model on the left of Fig. 1. , the second is logical business service showing as Controller, and the last one is view service. It's a logical gird, which is also called business grid in the Fig. 1. in fact. Each kind of resource is deployed as grid service. All services can be used and accessed through standard agreement, such as Simple Object Access Protocol (SOAP).

The process of business service with which GMM provides is like this:

1) When a client wants to access one business service through the browser. The interface which is provided by the company that provides the services of view (Service C) is given to client. The client input the demand through the interface, and then the view service requests the demand and disposes it to suitable operation or service which is provided by Service B.

2) The service B requests the demand from Service C and deals with it. If the demand relates to data services which is provided by Service A, it will send the demand to Service A for database operation. Otherwise, the demand will just be dealt with by the logical business service of Service B and the result will be sent back to Service C.

3) If Service A requests the demand from Service B, maybe from Service C which is not recommendable, the demand will be processed by the model service and the result will be sent back to client.

The whole model is according to concept of service, the service A/B/C might be provided only by a server, or by a local grid, even by a global grid. Then the grid services' providers can focus on the pivotal technique about their domain. And the services will be easy to change when they need changing.

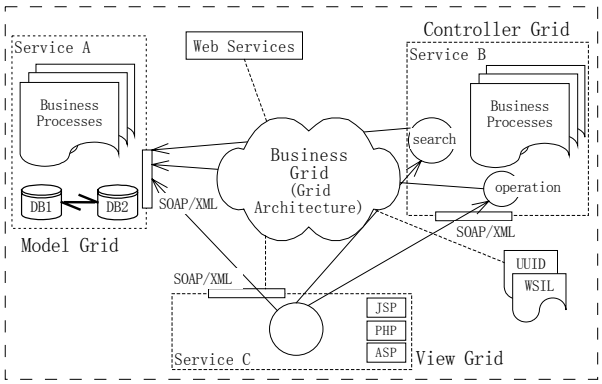


Fig. 1. Framework of GMM

Model Grid

The first kind of service which is deployed in Model Grid provides the services of operation on database. There are four tiers in the Fig. 2. The bottom tier is based on the core of Globus Toolkit. It provides the uniform interface of lifecycle, state management, service groups, factory, notification.

Supported by the uniform style of grid service interface, there is the second tier providing the data service of resource management, configuration, monitor management, etc, which is showed in the Fig. 2. And the basic grid database service is established at the foundation of second tier.

The SQL database service in the third tier is mainly used in the basic SQL operation of database.

Authorization and certificate service is concerned with establishing the identity of users or services (authentication), protecting communications, and determining who is allowed to perform what actions, as well as with supporting functions such as managing user credentials and maintaining group membership information.

Index services are mainly used in discovery operations. Basically, they provide a way to query and produce service data. Index services provide a client-side command called OGSI-find-service-data. The command allows you to query any service data element from any Grid service.

The top tier provides the OGSA architected services. They are basic operation of database, such as update, search, inserting and other expanding basic services. They should just be basic operation of unit database, while the complex operation for result set of data is processed at the grid of Controller.

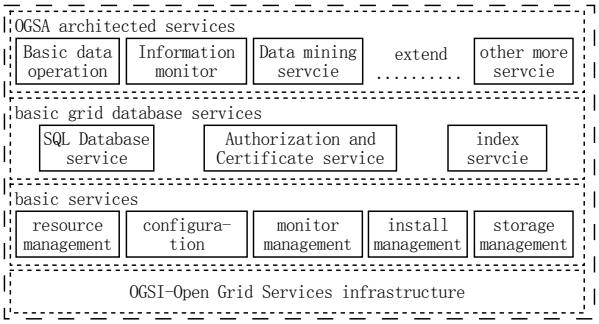


Fig. 2. Component of model grid

Controller Grid

Services of this part are built for processing the data from Model Grid or the View Grid. The bottom two tiers are almost the same with Model Grid. And the third tier is built on the bottom two and provides the basic logical controlling services. The most important part of logical service is the OGSA architected services, including the services of artificial intelligence, data mining, and other complex analysis of data. In addition, it provides the services of alternation, not refers to database, just between View Grid and Controller Grid.

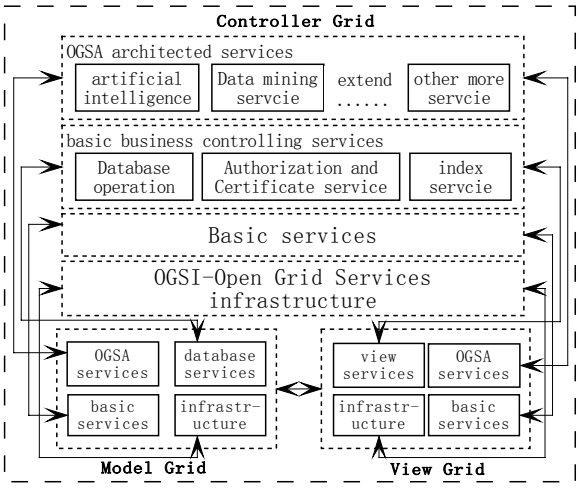


Fig. 3. Component of controller grid

View Grid

This kind of service is, comparing to other two, easier. Its mission is showing the result, which gets from Controller Grid to client. With different demands, there will be different interfaces. This domain is left for the team of art designing, interface designing and little knowledge of grid programming.

It provides interfaces for inputting the demands or parameters. Maybe JSP is chosen for View Grid service, or ASP, PHP, HTML, but no matter what kind of technique used, there is no processing for the demand or data in this area.

Application of GMM

There is an enterprise which has several servers distributed

in different places. When there is a client who wants some data from this enterprise, how to exploit the services based on GMM is as follows.

Model Service Design

There are five steps to achieve this service, which provides the basic SQL operation of database as follows:
1) Define the service's interface. This is done with GWSDL [9].

```
public interface SqlOperation{
   public void Operation(String StrSql);
   public databasetable getValue();}
```

2) Implement the service. This is done with Java.

```
public class SqlOperationImpl extends GridServiceImpl
implements SqlOperationPortType{
   public SqlOperationImpl(){}
   public void Operation(String StrSql){
   //execute SQL language and return result}
   public databasetable getValue() {}}
```

3) Define the deployment parameters. This is done with WSDD [3].
4) Compile everything and generate GAR file. This is done with Ant.
5) Deploy service. This is also done with Ant.

Logical Service Design

First, use the service of SqlOperation to get the data from data grid.

```
public class DO{
   public databasetable SqlOperation (String[] args){
   // Get command-line arguments
   // Get a reference to the SqlOperation
   // Call remote method "Operation"
   // Get current value through remote method}}
```

Then dealing with the data and providing a standard interface for View Grid. There are five steps the same to last section.
1) Define the service interface.

```
public interface BusinessProcess{
   public void Process(String StrSql){}
   public String[] getValue(){}}
```

2) Implement the service.

```
public class BusinessProcessImpl extends GridServiceImpl
implements BusinessProcessPortType{}
public void Operation(String Demand) throws
RemoteException {
   //execute search and return result to variable value
   value = DO.SqlOperation(StrSql);}
   //deal with the value According to the client demand
   public String[] DealValue(){}
   public String[] getValue() throws RemoteException{}}
```

3) Define the deployment parameters.
4) Compile everything and generate GAR file.
5) Deploy service.

Invoking of View Service

For providing the service of view, there is no need to exploit services of operation and processing about data, which have be exploited in the Model Grid and Controller Grid. As you can see in the Fig. 4. , there are various services provided by Controller Grid. What View Grid needs to do is just to know how to invoke the service and then invoke them. If the demand from View Grid involves the operation of data service, the controller service provided by Controller Grid will invokes remotely the data service through uniform interface defined by Model Grid. And the result that should have been dealt by the controller service will be returned to client according to the request of the client.

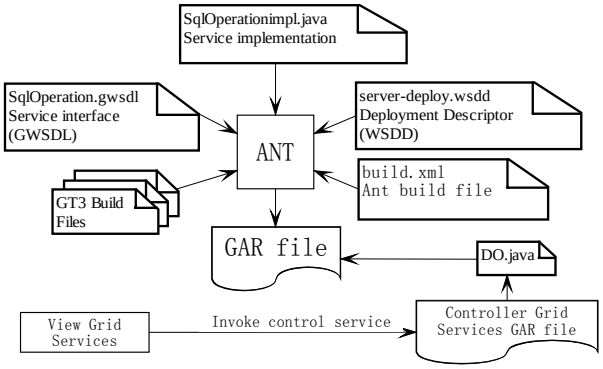


Fig. 4. The process of entire invoking

4. CONCLUSION AND FUTURE WORK

Developing a suit of service based on GMM is like this. Most services are more complicated than that. They will involve resource management, authorization and other components. This model is good for resolving the confusion of logic in the grid system. Some other advantages of this model are as follows:
1) The structure is clear. The model divides all functions into three parts; each part manages their own functions. It makes the function and structure clear at a glance.
2) The maintenance will be easy. With so clear structure and function processing, it is easier to maintain.
3) The using of resource is proper in this model. All resource is defined for service. All services are shared logically with others. And there is component, resource management, for the management of resource.
4) Achieving the share of resource in enterprise intranet and later extending to the scope of global.

This model is great for building of grid system, but there are several difficulties in it. The security is the one of most significant points and the management of resource is another one. With the development of technology, there will be a good resolve for them. Moreover, the united standard and norms are not easy to establishment. The intranet, especially in enterprise with multiplex and complicated business, really needs such a development model. But developing a global grid based this model, is not so easy indeed.

For all that, what should do next step is to apply this model to the transnational corporation which owning lots of servers distributed the entire world. And after resolve of the problems listed above, the application of this model will not be restricted at the intranet of enterprise.

5. REFERENCES

[1] F. Ian, K. Carl, et al, "The Anatomy of The Grid: Enabling Scalable Virtual Organizations", International journal of high performance computing applications, 2004,Vol.15 (4): 200-222.
[2] Y. Wu, D.H. Yuan, et al, "An Implementation Framework of Grid Computing Model Based on Globus Components", Journal of Sichuan University (Natural Science Edition), Aug 2005, Vol.42 (4):

[3] 719-724.
[4] Globus Toolkit 3.0 Quick Start, International technical support organization, Sep 2003, 23-35, 92-93.
[5] Alex Chaffee, What Is a Three-tier Architecture? http://www.jguru.com/faq/view.jsp?EID=125072.
[6] L. Ning, Z. Jian, et al, "Research on The Distributed System Based on EJB", Journal of Hefei University of Technology, Feb 2004, Vol.27(2): 221-224.
[7] J. Jin, L.F. Kong, et al, "The Installation and Configuration of Globus Toolkit3 in Grid Environment", Journal of the Hebei Academy of Sciences, June 2005,Vol.22 (2): pp.26-30.
[8] About the Globus Toolkit, http://www.globus.org/ toolkit /about.html.
[9] C. Karl, F. Steven et al, "Grid Information Services for Distributed Resource Sharing", Proc. 10th IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, 2001.
[10] Java programmer guide core framework, http://www. Unix.globus.org/toolkit/3.0/ogsa/docs/java_programmers_guide.html.

Parallel Search for Costas Arrays

Xinchun Yin, Tao Liu
Department of Computer Science and Engineering, Yangzhou University
Yangzhou, Jiangsu 225009, China
Email: xcyin@yzu.edu.cn, yzuisllt@sina.com

ABSTRACT

Costas arrays, special permutation matrices, have been applied to many fields such as signal processing and cryptography. However, so far the basic problem—the counting problem remains unsolved. Enumerating all costas arrays of order n from $n!$ permutation matrices has been of long standing interest. Here an effective parallel search algorithm is discussed. This algorithm adopts the classical master-slave technique. Master processor is in charge of allocating search subspace to slave processors. Every slave processor quests for costas arrays in assigned search subspace. Slave processors do not communicate with each other. And amount of communication between master and slave processors is small. So the speedup of this algorithm approaches total number of used processors.

Keywords: Costas Arrays, Counting Problem, Parallel Algorithm, and Master-slave Technique, Search.

1. INTRODUCTION

Costas arrays are widely applied to signal processing such as radar signal [1]. They can also be used in cryptography [2,3,4,5].

A costas array can be defined in several equivalent ways [6,7,8]. The difference triangle form is as follows.

Definition 1:

A costas sequence $\alpha_0, \ldots, \alpha_{n-1}$ is a sequence that is a permutation of the integers 1, …, n satisfying the property $\alpha_{s+k} - \alpha_s = \alpha_{t+k} - \alpha_t$ for every s, t and k, such that $0 \le s < t < t+k \le n-1$. An array that results from a costas sequence in this way is called a costas array.

Here, $\alpha_{s+k} - \alpha_s$ is a difference entry of order k.

(1 3 4 2 5) is an example costas sequence of length 5. Its difference triangle is shown below in Table 1.

Table 1. Costas sequence and its difference triangle

1	3	4	2	5
2	1	-2	3	
3	-1	1		
1	2			
4				

Corresponding costas array is

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

For convenience, we view costas sequence and costas array as the same concept.

2. GENERAL AND SEQUENTIAL SEARCH ALGORITHM FOR COSTAS ARRAYS—CASS

In [9] an observation about the difference triangle was made. Here we state and prove it as a lemma.

Lemma 1 For a permutation of length n, if there are no repeated entries in any row k of the difference triangle, $1 \le k < r \le n-1$, in row r at most r adjacent entries are different from each other.

Proof. In row r of difference triangle, we consider two arbitrary entries $\alpha_i - \alpha_{i-r}$ and $\alpha_j - \alpha_{j-r}$, $|j-i| < r$. For convenience let $j > i$.

$\alpha_i - \alpha_{i-r} = \alpha_i - \alpha_{j-r} + \alpha_{j-r} - \alpha_{i-r}$.

$\alpha_j - \alpha_{j-r} = \alpha_j - \alpha_i + \alpha_i - \alpha_{j-r}$. Since there are no repeated entries in any row k of the difference triangle, $1 \le k < r \le n-1$, we have $\alpha_{j-r} - \alpha_{i-r} \ne \alpha_j - \alpha_i$. Thus $\alpha_i - \alpha_{i-r} \ne \alpha_j - \alpha_{j-r}$. Varying i and j, in difference triangle we can see r adjacent entries of row r differ from each other.

Theorem 1 For a permutation of length n, whether it is a costas array can be determined. In the worst case, row 1 to $\lfloor (n-1)/2 \rfloor$ of difference triangle are checked for repeated entries in one row.

Proof. Let $f(k)$ denote the number of entries in row k of difference triangle, $1 \le k \le n-1$. We can see easily $f(k) = n-k$. Suppose row 1 to r-1 of difference triangle have been checked and no repeated entries in one row have been found. Now we can not determine whether the permutation is a costas array or not, and have to check row r. Based on Lemma 1, it is unnecessary to check row r when $f(r) \le r$. Just consider $f(r) > r$. That is, $n-r > r$, so maximum of r takes $\lfloor (n-1)/2 \rfloor$.

Basic idea of sequential search algorithm for costas arrays is as follows. Starting from a permutation of length n, all permutations of length n can be generated by swapping entries of the permutation. In such process, test can be made for costas arrays. Suppose sometime the permutation is α_0, $\alpha_1 \ldots \alpha_{m-1}$, $\alpha_m \ldots \alpha_j \ldots \alpha_{n-1}$. Here α_0, $\alpha_1 \ldots \alpha_{m-1}$ called initial costas array satisfies costas array condition. Pointer j points to α_j being dealed with, and pointer m points to the end of current initial costas array. If α_0, $\alpha_1 \ldots \alpha_{m-1}$, α_j forms a new initial costas array, exchange α_j and α_m when $j > m$, and add one to m. Otherwise, add one to j, which means pruning.

Let us define some data structure.

diff[1..n/2-1][1..2×n-1]. *diff*[k][t] takes 0 or 1, $1 \le k \le n/2-1$, $1 \le t \le 2 \times n-1$. *diff*[k][t]=1 shows that for a permutation row k of difference triangle contains entry t-n.

ubs[0..n-1]. After initial costas array α_0, α_1, ..., α_{m-1}, α_m is determined, difference entries like $\alpha_m - \alpha_{m-k}$

must be stored for further search, $1 \le k \le ubs[m]$.

$ubj[0..n-1]$. When checking whether α_j is feasible so that α_0, α_1, ..., α_{m-1}, α_j forms a new initial costas array, $\alpha_j - \alpha_{j-k}$ needs to be verified to see whether it has existed in row k of difference triangle for α_0, α_1, ..., α_{m-1}, $1 \le k \le ubj[m]$.

$ubc[0..n-1]$. Considering α_j, $\alpha_j - \alpha_{j-k}$ requires to be computed, $1 \le k \le ubc[m]$.

Now general and sequential search algorithm for costas arrays can be described as follows.

```
Algorithm CostasArraySequentialSearch(CASS)
Input: order n of costas arrays to be searched for, initial
       costas array costas[0..INILEN-1] of length INILEN
Output: all costas arrays of length n beginning with
       costas[0..INILEN-1]
Begin
  Extend initial costas array costas[0..INILEN-1] to a
  permutation of length n beginning with costas;
  mid:=(n+1) div 2;  m:=INILEN;  j:=INILEN;
  while m≥INILEN do
    while j<n do
      for i:=1 to ubc[m] do
        d[i]:=costas[j]-costas[m-i];
        if i≤ubj[m] and diff[i][n+d[i]]=1 then
          break
        end if
      end for
      if i≤ubc[m] then j:=j+1; continue end if
      if j>m then costas[j]↔costas[m] end if
      if costas[0]>mid then halt end if
      if m=n-1 then
        store the costas array costas[i];
        if costas[0]<mid or (costas[0]=mid and n is even)
        then store the costas array n+1-costas[i] end if
      end if
      for i:=1 to ubs[m] do diff[i][n+d[i]]:=1 end for
      m:=m+1;
      push(j);  //put j into the stack
      j:=m;
    end while
    m:=m-1;
    if m<INILEN then halt end if
    for i:=1 to ubs[m] do
      diff[i][n+costas[m]-costas[m-i]]:=0
    end for
    j:=pop;  //pop data from the stack
    if j>m then costas[j]↔costas[m] end if
    j:=j+1;
  end while
End
```

On the value of $ubs[i]$, $ubj[i]$, and $ubc[i]$, there are following three theorems.

Theorem 2 $ubs[i]=\begin{cases} i, & i \le \lfloor (n-1)/2 \rfloor \\ n-1-i, & i > \lfloor (n-1)/2 \rfloor \end{cases}$

Proof. When $i \le \lfloor (n-1)/2 \rfloor$, in near future it is necessary to check whether difference entries $\alpha_i - \alpha_0$ and $\alpha_{2i} - \alpha_i$ are different, so $ubs[i]=i$. When $i > \lfloor (n-1)/2 \rfloor$, in the end it is necessary to check whether difference entries $\alpha_{n-1} - \alpha_i$ and $\alpha_i - \alpha_{2i+1-n}$ are different, so $ubs[i]=n-1-i$.

Theorem 3 $ubj[i]=\lfloor i/2 \rfloor$.

Proof. For a permutation the first entry of row r of difference triangle is $\alpha_r - \alpha_0$. From Lemma 1, it is necessary to check whether $\alpha_i - \alpha_0$ is the same as $\alpha_r - \alpha_0$. Here $i-r \ge r$. So $r \le \lfloor i/2 \rfloor$, and $ubj[i]=\lfloor i/2 \rfloor$.

Theorem 4 $ubc[i]=\max(ubs[i], ubj[i])$.

Proof. Computing difference is just for two purposes: storing difference for further search, and checking the repetition of difference. So $ubc[i]=\max(ubs[i], ubj[i])$ naturally.

Previous algorithms have applied Theorem 1 [1,10,11,12,13], while CASS applies Theorem 4 and thus reduces the amount of computation of difference entries.

3. GENERAL AND PARALLEL SEARCH ALGORITHM FOR COSTAS ARRAYS—CAPS

To achieve parallelism in algorithm CASS, we use the classical master-slave technique [14]. The master generates initial costas arrays of fixed length *INILEN* for some *INILEN*. The slave voluntarily asks the master for search task. Then the master passes an initial costas array of length *INILEN* to the idle slave, who in turn continues to search for costas arrays that contain the fixed prefix of length *INILEN*. The slave returns all found costas arrays to the master and then asks for another initial costas array of length *INILEN*. The master continues to compute initial costas arrays of length *INILEN* until no more initial arrays beginning with i exist, $1 \le i \le \lceil n/2 \rceil$, due to symmetry of costas arrays.

```
Algorithm CostasArrayParallelSearch(CAPS)
Input: order n of costas arrays to be searched for
Output: all costas arrays of order n
Master processor algorithm
procedure CAPSMaster
Begin
  active_slaves:=p;
  while active_slaves>0 do
    receive signal signal from processor i;
    if signal=Accomplished then
      receive and record costas arrays found by processor i
    end if
    if has_initial_costas_arrays then
      send processor i signal NewTask;
      send processor i an initial costas array
    else
      send processor i signal Terminate;
      active_slaves:=active_slaves-1
    end if
  end while
End
Slave processor algorithm
procedure CAPSSlave
Begin
  Send master processor signal Ready;
  finished:=false;
  While not finished do
    Receive signal signal from master processor;
    if signal=NewTask then
      Receive a new initial costas array from master;
      Execute algorithm CASS;
      Send all found costas arrays to master processor
    else //signal=Terminate
      finished:=true
    end if
  end while
End
```

4. IMPROVEMENT ON CAPS FOR SPECIAL CASE

Suppose costas arrays of order *n* need to be enumerated and costas arrays for orders less than *n* have been gotten. Initial costas arrays beginning with 1, or (2 1), or (2 3 1), or (2 *n* 1) and so on, can be inserted into a known smaller-sized costas array and tested more efficiently. Such initial costas arrays will not be allocated to slave processors. What to do is just adding upper corner dots to smaller costas arrays and testing them simply and easily.

5. IMPLEMENTATION OF CAPS

We implement CAPS with C and MPI in Shengteng1800 of Lenovo. For small orders, we verify the number of costas arrays counted by previous work. However, faced with rough *n*! permutations, enumeration of costas arrays for orders higher than 23 is a huge task. For example, it takes nearly 30 years of CPU time to enumerate all costas arrays of order 26 [13].

Due to computational resource, we are unable to get new enumeration results. Here we merely discuss the design of search algorithm theoretically. But our algorithm is more efficient with the same computational resource using Theorem 4.

6. REFERENCES

[1] J.K.Beard, J.C.Russo, et al, "Combinatoric collaboration on Costas arrays and radar applications", Proceedings of the IEEE Radar Conference, 2004.

[2] X.C.Yin, T.Liu, "On the Existence and Counting Problems of Costas Arrays and its Signature Application", Proceedings of 40th Conference on Information Sciences and Systems, 2006.

[3] M. Sterling, E. L.Titlebaum, et al, "An Adaptive Spread-Spectrum Data Hiding Technique for Digital Audio", Proceedings of IEEE ICASSP, 2005.

[4] J.Q.Ouyang, R.R.Liu, et al, "Some algebraic properties of Costas arrays", Microelectronics & Computer, No.10, 2003, 83-88.

[5] P.A.Gao, G.Y.Bu, et al, "On Costas arrays", Computing Technology and automation, 2000,19(3):10-15.

[6] S.Rickard, "Searching for Costas arrays using periodicity properties", Proceedings of the IMA 6th International Conference on Mathematics in Signal Processing, 2004.

[7] S.W.Golomb, O.Moreno, "On periodicity properties of Costas arrays and a conjecture on permutation polynomials," IEEE IT, 1996,42(6): 2252-2253.

[8] S.W.Golomb, H.Taylor, "Constructions and properties of Costas arrays," Proc.IEEE,72(9), 1984, 1143-1163.

[9] J.Silverman, V. E. Vickers, et al, "On the number of Costas arrays as a function of array size," Proceedings of the IEEE,1988,76(7).851-853.

[10] O.Moreno, J.Ramirez, Dorothy Bollman et al, "Faster backtracking algorithms for the generation of symmetry-invariant permutations", Journal of Applied Mathematics, 2002, 2(6). 277-287.

[11] S.Tang, J.Y.Zhou, "The general search algorithm for Costas arrays based on trimming method", Natural science Journal of Xiangtan University, 2000,22(4): 31-34.

[12] MacTech, "December 1999 Programmer's Challenge", available on web page http://www.mactech.com/progchallenge/9912Challeng.html.

[13] S.Rickard, E. Connell, et al, "The Enumeration of Costas Arrays of Size 26", Proceedings of 40th Conference on Information Sciences and Systems, 2006.

[14] G.L.Chen, H.An, et al, Practice of Parallel algorithm, Beijing: Higher Education Press, 2004.

,

Parallel Ant Colony Algorithm for Multiprocessor Scheduling with Communication Delays

Xiaohong Kong1,2
1School of Information Technology, Southern Yangtze University, Wuxi, Jiangsu, 214122, China
2 Henan Institute of Science and Technology, Xinxiang, Henan 453003, China
Email:nancykong@hist.edu.cn
Jun Sun
School of Information Technology, Southern Yangtze University, Wuxi, Jiangsu, 214122, China
Email:sunjun_wx@hotmail.com
Wenbo Xu
School of Information Technology, Southern Yangtze University, Wuxi, Jiangsu 214122, China
Email:xwb@sytu.edu.cn

ABSTRACT

Generally speaking, tasks scheduling in multiprocessor systems is NP-hard even if under strictly simplifying assumptions. In this paper, we develop a parallel ACO to solve the multiprocessor scheduling on distributed memory architecture. Based on message passing interface, multiple sub-ant-colonies evolve respectively and interchange the information every fixed k iteration to enhance the search ability of algorithm. The experiment results show that the proposed algorithm performs better in solution quality as well as in scalability.

Keywords: Ant colony algorithm, DAG, Multiprocessor scheduling, List scheduling, parallel algorithm.

1. INTRODUCTION

Many combinatorial optimization problems are known as NP completes even NP hard. There are no efficient algorithms that can solve the problems with polynomial time complexity. Metaheuristics are widely employed methods to generated optimal or sub-optimal solutions for the problems with tolerable time consumption. For large-scaled combinatorial optimization problem, however, the computation times associated for metaheuristics to find approximate solutions may be very large and the complexity is high. Due to the advent of fast communication networks and the availability of computing resources, a natural trend is parallelizing the metaheuristics to speed up the exploration of the solution space for approximate solutions of combinatorial optimization problems. In contrast to their sequential counterparts, parallel implementations of metaheuristics have better performance in both finding improved solutions and robust using the impact of the cooperation between agents [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. In this paper, we will discuss a well-known class of combinatorial problems, task-scheduling (TS) problems on distributed multiprocessor system. We employ a Parallelized Ant Colony (PACO) algorithm to solve the scheduling problems effectively and efficiently. The experiment results that PACO is a promising problem solver for TS problems.

2. PROBLEM STATEMENTS

When a parallel application is executing in a distributed multiprocessor systems, it is usually be decomposed into subtasks, which have diverse computation requires, an important challenge in this filed is the matching and scheduling of subtasks. Matching tackles with the problem selecting suitable machines for the subtasks in order to achieve the high performance goal using some mapping strategies. Scheduling, including task scheduling and message scheduling, determines the execution order of subtasks on same machines and computes the starting time and the finishing time. Inappropriate scheduling of tasks can fail to exploit the full potential of a distributed system due to significant overhead if interprocessor communication (IPC) was considered, and can offset the gains from parallelism.

Multiprocessor scheduling problem (MSP) is popularly modeled by a weighted directed acyclic graph (DAG) or micro-dataflow graph, and the objective of MSP is minimizing the parallel completion time or schedule length by properly assigning the nodes of the graph to the processors without violating the precedence constraints. Generally, the multiprocessor scheduling problem is NP-hard [1] except for some cases in which an optimal solution can be obtained in polynomial time. In the past decades, a myriad of heuristic algorithms have been investigated [2, 3], but the results are constrained by efficiency and complexity.

A directed acyclic graph (DAG) can be represented by a tuple $G = (V, E, C, W)$, where V is a set of V nodes and E is a set of E edges, corresponding to the data dependence among tasks. Each task is assumed to have an estimated execution time $W(n_i)$, denoting the amount of work associated with task $n_i \in V$. The edge $e_{i,j} = (n_i, n_j) \in E$ is associated with an estimated communication cost $c(n_i, n_j)$, representing the amount of data units to be transferred from n_i to n_j. The edge also represents the partial order between tasks n_i and n_j, which dictate that task n_j cannot be executed unless all its predecessors n_i have been completed their execution. If two tasks are scheduled to the same processor, the communication cost between them is assumed to be negligible. The sets of immediate predecessor and successors of n_i are given by $prce(n_i)$ and $succ(n_i)$ respectively. The target system is commonly

assumed to consist of p processor connected by an interconnected network based on a certain topology in which a message is transmitted through bidirectional links with the same speed. A fully connected network, a hypercube or mesh is often discussed as the model.

3. ANT COLONY ALGORITHM

The ACO was introduced by Dorigo et al. [11] inspired from the behavior of natural ants foraging food. It is a population-based evolutionary approach where individuals of artificial ants share the path information and search for good solutions throughout problem space. The ACO has been successfully applied to combinatorial optimization problem utilizing the swarm intelligence of a colony of ants [4, 5, 6, 7, 11].

One of the key themes of the researches of the ant colony system is how to balancing exploitation of the previous solutions and exploration of the search space so that the algorithm finds a compromise between a local minima and fast convergence. According to [11], the initial framework of the ACO applied to Traveling Salesman Problem (TSP) works as follows: m ants must traverse a given set of n cities and ensure that every city is visited exactly once while minimizing objective function. Every ant of generations constructs a solution independently by deciding which node is selected as the next node according to state transition rule step by step until a tour is finished. When Ants search a tour, they deposit pheromone on their paths and also modify the amount of pheromone on the visited edges by applying the local updating rule. Finally, applying the global updating rule modify the amount of pheromone on edges again. The following ants of the next generation incorporate the pheromone deposited in paths and a specified-problem heuristics and so forth.

Let d_{ij} be the distance between city i and city j and τ_{ij} the amount of pheromone in the edge that connects i and j. An ant located at city i selects an edge between city i and city j according to the probability based on the density of pheromone trail τ_{ij} and a heuristic value η_{ij}, where the heuristic value $\eta_{ij} = 1/d_{ij}$ is generally used. Both heuristic information and pheromone information decide: An edge with a high amount of pheromone is a very desirable choice [11].

3.1 ACS State Transition Rule

In ACS the state transition rule is as follows given by Eq. (1) (2), an ant positioned on node r chooses the next city s to move to. Where q is a random number uniformly distributed in [0, 1], q_0 is a parameter ($0 < q_0 < 1$), and S is a random variable selected according to the probability distribution given in Eq. (2).

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{[\tau(r,u)][\eta(r,u)]^{\beta}\} & \text{if } q \le q_0 \\ S & \text{otherwise} \end{cases} \tag{1}$$

$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)][\eta(r,s)]^{\beta}}{\sum_{u \in J_k(r)} [\tau(r,s)][\eta(r,s)]^{\beta}} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The state transition rule resulted from Eqs. (1) and (2) is called pseudo-random-proportional rule. This state transition rule favors transitions towards nodes connected by short edges and with a large amount of pheromone. The parameter q_0 determines the relative importance of exploitation versus exploration: Every time an ant in city r has to choose a city s to move to, it samples a random number $0 < q < 1$. If $q < q_0$ then the best edge according to Eq. (1) is chosen (exploitation), otherwise an edge is chosen according to Eq. (2) (exploration).

3.2 ACS Local Updating Rule

While building a solution of the TSP, ants visit edges and change their pheromone level by applying the local updating rule of Eq. (3)

$$\tau(r,s) \leftarrow (1-\alpha)\tau(r,s) + \alpha\Delta\tau(r,s) \tag{3}$$

Where $0 < \alpha < 1$ is a pheromone decay parameter. $\Delta\tau(r,s)$ has three choices, and reader can refer to literature [1].

3.3 ACS Global Updating Rule

Global updating is performed after all ants have completed their tours. In ACS only the globally shortest tour from the beginning of the trial is allowed to update the pheromone. This choice direct ants search in a neighborhood of the best tour found up to the current iteration of the algorithm. The global pheromone level is updated according to Eq. (4).

$$\tau(r,s) \leftarrow (1-\gamma)\tau(r,s) + \gamma\Delta\tau(r,s) \tag{4}$$

Where,

$$\Delta\tau(r,s) = \begin{cases} Q/L & \text{if } (r,s) \in \text{global-best-tour} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$0 < \gamma < 1$ is the pheromone decay parameter, and L is the length of the globally best tour from the beginning of the trial. Eq. (5) dictates that only those edges belonging to the globally best tour will receive reinforcement. The author also tested another type of global updating rule, called iteration-best, as opposed to the above called global-best, which use the length of the best tour in the current iteration of the trial, in Eq. (5) [11].

4. PARALLEL ANT ALGORITHM FOR MULTIPROCESSOR SCHEDULING

Most parallel implementations of ant algorithms differ only in granularity and in whether the colonies exchange information are done locally in all colonies or centrally by a master processor [4].

4.1 Related Works

There are a few parallel ant algorithms for implementations in the literature. Stützle [4] discussed two versions for

parallelization: the independent run of multiple copies of the same algorithm, and the speed up of a single copy of the algorithm based on a message passing architecture. In the first case, every processor can has one different set of parameters and they have the advantage to be easily run in parallel. The second case used master-slave architecture to keep the pheromone trail and update it with a local search procedure.

Pierre Delisle et al. [5] designed a parallel ant colony system using share-memory architecture. The objective of this paper is to show that an internal parallelization of relatively fine grain can yield good performance.

Talbi et al. [6] have proposed a powerful and robust algorithm with a local search procedure based on tabu search to solve the Quadratic Assignment problem. They used a fine-grained master-worker approach, the master computes the new pheromone matrix and sends it to the workers, and every worker holds a single ant that produces one solution. Every worker then sends its solution to the master.

Middendorf et al. [7] studied four strategies for the exchange of information among multiple ant colonies. They show that Instead of exchanging the local best solution very often and between all colonies it is better to exchange the local best solution only with the neighbor in a directed ring and not too often.

Shu-Chun Chu et al. [9] first partitioned the artificial ants into several groups, and then developed seven communication methods for updating the pheromone level between groups in ACS.

Bullnheimer et al. [10] propose two parallel algorithms: a Synchronous algorithm and a Partially Asynchronous Parallel Algorithm. Their approach of the parallel ant type is an internal parallelization composed of a master processor and multiple slave processors.

4.2 independent runs of ant colony algorithm

The simplest method is independent runs of one algorithm simultaneously without interchanging information with each other. Each process is assigned to a physical processor, and these process works as a sequential process [7]. Every processor can select a set of parameter respectively.

4.3 Multiple Sub-ant Colonies with Communication

In the distributed memory parallel machines architecture, MPI is a proposal for the standardization of a message-passing interface, with the aim of enabling program portability among different parallel machines. Each of these computers executes a process used to control the exchange of messages among them. This Parallel Processing Elements (PPE) on which the algorithms are executed is different from the Processing Elements (PE) in the target system [3].

Here, we use the best solution in the current iteration of the trial and the globally best solution from the beginning of the trial in every sub-ant colony as the exchange information in order to enhance the diversity of solution space. A master processor collects the global pheromone and broadcasts the best solution to other processors, and every processor constructs the solution and updates the locally pheromone. When a colony receives a solution that is better than the best solution in the current iteration by this colony, the received solution becomes the best solution in the current iteration. Then every colony computes the new pheromone information, which is usually stored in local memory. It influences the colony because during trail update some pheromone is always put on the trail that corresponds to the best solution in the current iteration.

4.4 Parallel Ant Colony Algorithm for Multiprocessor Scheduling

In literature [12], the authors state that a coarse-grained parallelization of the ACO is more promising than a fine-grained parallelization because of the high communication costs engendered by the management of the pheromone. Using a MPI model, we develop a parallel ACO algorithm to solve the multiprocessor scheduling in distributed memory architecture. Every processor can hold a colony of ants and after every several generations the colonies exchange information about their solutions, similar to the principles in the island model of genetic algorithms [8].

We incorporate the parallel ant colony algorithm with list scheduled for multiprocessors using the ant colony to produce a task list according to some priority rule. The node priorities in this paper are determined by cooperation among the multiple ant colonies, and then these tasks are assigned to different processors. When we design the parallel ant colony to construct the solution, pheromone trail is deposited in path and the task SL is utilized as heuristics to dynamically selects the pair of the ready task and the match processor. In every information exchange step, evaluating the scheduling length and the best solution is sent to all colonies where it becomes the new best solution and updates the information.

5. RESULTS

5.1 Parameter Selection

The performance of stochastic algorithm depends crucially on diverse parameters, so we must select favorite parameter to improve solutions. We denote the population size, fixed interval cycle, the max iteration number as NP, Kcycle, MAXiter respectively. Here, Kcycle =3, 5, 10, NP=20, 30, MAXiter =30, 50, Q= 100, 1000 respectively.

5.2 Qualities of Solutions

Benchmark instances from website (http://www.mi.sanu.ac.yu/ tanjad/) [14] are used to test the validity of our approach with different processors size.

Table1. Results of the parallel ACO with different processors compared against optimal solutions (% deviations)

graph	No. of PPEs				
graph	1	2	3	4	5
ogra50	0.00	0.00	0.00	0.00	0.00
ogra100	0.00	0.00	0.00	0.00	0.00
ogra150	10.50	8.70	6.70	5.00	7.40
ogra200	11.33	8.25	9.33	9.58	6.83
ogra250	15.71	13.5	12.00	11.79	12.86
ogra300	14.19	13.56	11.69	12.19	12.38
ogra350	13.47	12.78	11.72	11.33	12.25
Avg. Dev.	9.31	8.11	7.35	7.13	7.38

In our experiments, we select the ogra50-ogra350 problem with the same density 90 and target machine size 4. Here density is defined as the number of edges divided by the number of edges of a completely connected graph with the same number of nodes [13]. For each task size, the deviations of the best solution from the optimal schedule for 10 runs are recorded in Table1. Parallel ACO always has comparable performance on the results of sequential counterpart. Here, Kcycle=5, NP=20, MAXiter=30.

5.2 The Performance of Information Exchange

The ant colony algorithm is applied to each processor and communication between processors every some fixed cycles. We emphasize impact of the cooperation between agents for information exchange differing in the degree of coupling and frequency that are performed between the colonies through this exchange. We select Kcycle=3, 5, 10 as fixed cycle to swap the best solutions to the benchmark problem RGNOS (r1p.50.10.0) from website [15].

The result is shown in Tab.2 is the best solution out of 10 runs to the different task size without known optimal solutions. It is evident that the results are concerned with processor number, the fixed circles, and task number. When PPE is more, a less fixed cycle is appreciated.

Table 2. The best solution of the parallel ACO with different fixed cycle

Graph r1p.50.10.0	No. of PPEs											
	Kcycle =3				Kcycle= 5				Kcycle =10			
	2	3	4	5	2	3	4	5	2	3	4	5
100	3315	3444	3404	3403	3324	3353	3369	3417	3427	3368	3313	3352
150	3891	3906	3949	3925	3958	3876	3953	3874	3902	3905	3924	3873
200	7170	7240	7162	7132	7248	7204	7131	7143	7240	7183	7236	7191
250	5475	5550	5535	5540	5498	5502	5517	5520	5545	5538	5497	5470
300	7881	7924	7935	7940	7899	7901	7921	7918	7938	7935	7920	7880
350	11220	11234	11250	11238	11226	11236	11230	11236	11268	11239	11230	11230
averge	6492	6549.7	6539.2	6529.7	6525.5	6512	6520.2	6518	6553.3	6528	6520	6499.3

In order to evaluate the gain brought by the ant cooperation system, we have also compared the results of the two-shared information: the global best solution and the current best solution. Table 3 is the deviations of the best solution from the optimal schedule for 10 runs. The global best information is slightly preference than the current generation best information.

Table 3. Results of the parallel ACO with different information exchange compared against optimal solutions (% deviations)

graph	No. of PPEs							
	Current best solution				Global best solution			
	2	3	4	5	2	3	4	5
ogra50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ogra100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ogra150	7.50	8.10	5.30	6.40	7.10	6.30	6.40	6.40
ogra200	9.25	9.58	9.33	6.83	7.75	7.67	9.08	6.75
ogra250	10.5	7.50	9.20	5.10	9.20	6.80	5.90	8.04
ogra300	11.6	9.00	10.1	6.00	10.35	9.20	6.10	6.80
ogra350	15.23	12.53	11.01	12.83	13.97	12.05	12.21	11.72
Avg. Dev.	7.73	6.67	6.42	5.31	6.91	6.01	5.65	5.69

6. CONCLUSIONS

In this paper, we employ a novel population-base search technique, Ant Colony Optimization (ACO), in list scheduling and propose a parallel ACO algorithm for tasks scheduling. Ant Colony Optimization, in our proposed algorithm, is used to obtain a scheduling list of nodes whose schedule length through afterwards task allocation is minimal.

7. EFERENCES

[1] Y.K. Kwok, I. Ahmad, "Efficient Scheduling of Arbitrary Task Graphs to Multiprocessors Using a Parallel Genetic Algorithm", Journal of Parallel and Distributed Computing, 1997, vol.47: 58-77.

[2] T.M. Nabhan, A.Y. Zomaya, "A Parallel Simulated Annealing Algorithm with Low Communication Overhead", and IEEE Transactions on Parallel and distributed Systems, Dec. 1995, Vol.6: 1226-1233.

[3] Y.K. Kwok, I. Ahmad, "A Parallel Algorithm for Compile-Time Scheduling of Parallel Programs on Multiprocessors", Proceedings of the 1997 International Conference on Parallel Architectures and Compilation Techniques, 1997, 90-101.

[4] T.Stutzle, "Parallelization strategies for ant colony optimization", Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, Lecture Notes in Computer Sciences, vol. 1498, 1998, 722-731.

[5] P. Delisle, M. Gravel, et al, "A shared memory parallel implementation of ant colony optimization", Proceedings of the 6th Metaheuristics International Conference (MIC'2005), Vienne, 2005, 257-264.

[6] E-G. Talbi, O. Roux, et al, "Parallel ant colonies for combinatorial optimization problems", Proceedings of the 11 IPPS/SPDP'99 Workshops, Lecture Notes in Computer Sciences, vol. 1586, Springer-Verlag, 1999, 239-247.

[7] M. Middendorf, F. Reischle, et al, "Information Exchange in Multi Colony Ant Algorithms", Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing, Lecture Notes in Computer Sciences, vol.1800, Springer-Verlag, 2000, 645-652.

[8] A. L. Corcoran, R. L. Wainwright, "A parallel island model genetic algorithm for the multiprocessor scheduling problem", Proceedings of the 1994 ACM/SIGAPP Symposium on Applied Computing, 1994, 483-487.

[9] S.C. Chu, John F. Roddick, et al, "Ant colony system with communication strategies", Information Sciences, 2004, vol.167: 63-76.

[10] B. Bullnheimer, G. Kotsis, et al, "Parallelization Strategies for the Ant System", Technical Report POM 9/97, University of Vienna, 1997.

[11] M. Dorigo, V. Maniezzo, et al, "The Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man, and Cybernetics--Part B, vol. 26, No. 2, 1996, 29-41.

[12] M. Dorigo, T. Stützle, "The Ant Colony Optimization Metaheuristic: algorithms, applications and advances", Handbook of Metaheuristics, F. Glover and G. Kochenberger (eds), Boston, MA, Kluwer, 2003, 251-285.

[13] T. Davidovic, T.G. Crainic, "Benchmark-problem instances for static scheduling of task graphs with communication delays on homogeneous multiprocessor systems", Computers and Operations Research, vol. 33, No. 8, 2006, 2155-2177.

[14] http://www.mi.sanu.ac.yu/ tanjad/.

[15] http://www.eee.hku.hk/~ykwok/publications.html.

An Improving Computation and Data Decomposition Method

Chunli Dong，Rongcai Zhao，Jianmin Pang，Lin Han，Peng Du
National Digital Switching System Engineering & Technology R &D Center
Zhengzhou 450002, P.R CHINA
Email: dclldh@sina.com

ABSTRACT

Data locality is critical to achieving high performance on high-performance parallel machine. So how to find a good data decomposition is becoming a key issue in parallelizing compilers. We have developed a compiler system that fully automatically parallelizes sequential programs and optimizes data decomposition to improve data locality. Our data decomposition algorithm consists of two steps. The first step chooses the basic data and computation decomposition without considering read-only data. The second step then changes the data decomposition considering read-only data. We ran our compiler on a set of application programs. The results show that the algorithm can effectively discovers parallelism.

KEYWORDS: data decomposition, computation decomposition, read-only data, loop level parallelism.

1. INTRODUCTION

With the development of high-performance parallel machines [1], the massive parallel computers provide a cost-effective scalability to solve many large scale scientific problems, but these systems are very difficult to program and use. Although the availability of programming models such as High Performance Fortran [2] offers a significant step towards making these systems truly usable, the programmer is forced to design parallelization and data mapping strategies which are heavily dependent on the underlying system characteristics. Efficient utilization of parallel machines requires writing parallel programs or designing parallelizing compilers that can convert existing sequential programs to parallel code. So the research of automatic parallel compilation is becoming a key issue in this field. Finding a good decomposition of computation and data, minimizing communication by increasing the locality of data references is an important optimization for achieving high performance. We have developed a compiler system that can fully automatically parallelize sequential programs.

2. RELATED WORKS

A popular approach to the decomposition problem is to solicit the programmer's help in determining the data decomposition, such as Fortran D [3], Vienna Fortran [4] and HPF [5]. Several alternative data parallel languages also have been proposed. One example is the Fx project [6] at Carnegie Mellon University that has proposed a new dialect of HPF that enables the programmer to specify both data parallelism and task parallelism in order to handle a wider range of scientific problems than ordinary data parallel languages. SUIF [7] and Parafrase-2 [8] are typical examples of conventional parallelizing compilers that mainly target bus-based shared memory architectures. We extend SUIF to retarget distributed memory machines and generate message-passing code using MPI (message passing interface) in our compiler system.

In this paper we propose a framework that can automatically finds computation and data decomposition, in order to translate more serial programs into parallel programs, we improve the decomposition algorithm with read-only data recognition that can reduces data reorganization communication. The rest of the paper is organized as follows. In section 3 we describe the mathematical framework of decomposition method, in section 4 we describe a linear data and computation decomposition method with read-only data recognition, in section 5 we analyze a simple example using the data and computation decomposition method, and we conclude in section 6.

3. THE MATHEMATICAL FRAMEWORK OF THE DECOMPOSITION METHOD

3.1 Problem Statement

In distributed memory mode parallel machines, techniques for maximizing parallelism and locality within a single loop nest have been presented in the literature [9]. A number of researchers have also looked at the specific problem of mapping a single loop nest onto parallel machines [10]. But how to find an optimizing data and computation decomposition across multiple loop nests is still an unsolved problem.

In example 1, if we consider each loop nest individually, there is no dependence in loop nest L1, the outermost and innermost loop are all do-all loop, array A and B may be distributed by column or row. In the second loop nest, there is dependence in the outermost loop, array B must be distributed by column and array C must be distributed by row. So if we consider the two loop nests together, the final distribution is array A and B be distributed by column, array C be distributed by row.

```
Example 1:  L1:  For i=0 to N
                   For j=0 to N
                     A[i][j]+= B[i][j];
            L2:  For i=0 to N
                   For j=0 to N
                     C[j][i]=C[j][i-1]+B[i][j-1];
```

The problem of optimal distribution is a NP-Complete problem, this was proved by Kennedy and Kremer [11], but many researchers have given some distribution methods for special program model. The decomposition method we discussed in this paper is loop level parallelism, the loop bounds and array subscripts are

affine functions of the loop indices and symbolic constants, and the number of iterations is much larger than the number of processors. The key issues we discuss are how to formulate the system of equations, and then how to solve the questions efficiently.

3.2 Mathematical Framework

For distributed memory machines, loop level parallelism is implemented by the iteration space partition. We discuss this problem in two steps. First, the computation and data are mapped onto virtual processor space. The virtual processor space has as many processors as needed to fit the number of loop iterations and the sizes of the arrays. Second, the processors of the virtual processor space are mapped onto the physical processors of the target machine. In this paper, we discuss the first step mainly.

1) Form of Loop Nest

When we analyses the problem, we represent data and computation decomposition as affine transformations. In this case, all loops are normalized to have a unit step size, and all arrays subscripts are adjusted to start at zero. The loop bounds and array subscripts are affine functions of the loop indices and symbolic constants.

```
L1:     For i=0 to N
            For j=0 to N
                X[i][j]=X[i][j]+...X[i+m][j+m]+Y[i][j]+...;
```

In this example, all arrays subscripts $i_{m*n} = f_{mn}(i, j)$ $j_{m*n} = g_{mn}(i, j)$ are affine functions of the loop indices.

2) The Basic Concepts

When we build up the mathematical model, the vector spaces we needed are as follows:

An iteration space I : A loop nest of depth l defines an iteration space I , each iteration of the loop nest is identified by its index vector $\vec{i} = (i_1, i_2, ...i_l)$.

An array space A : an array of dimension m defines an array space A , and each element in the array is accessed by an index vector $\vec{a} = (a_1, a_2, ...a_m)$.

A processor space P : a n dimension processor array defines a processor space P , a n dimension rectangle.

Based on the three vector spaces, we can give the following three definitions.

Definition 1: The reference between l dimension iteration space and m dimension array space is represented by the affine function of array access:

$I -> A: f(\vec{i}) = F\vec{i} + \vec{k}$, F is a $l*n$ linear transform matrix, $\vec{k}$ is a constant vector.

Definition 2: Let $\vec{a} = (a_1, a_2, ...a_m)$ be an index vector for a m dimension array. The affine data decomposition of the array onto a n dimension processor space is an affine function $\vec{d}$:

$A -> P, \vec{d}(\vec{a}) = D\vec{a} + \vec{\delta}$, D is a $n*m$ linear transformation matrix and $\vec{\delta}$ is a constant vector.

Definition 3: Let $\vec{i} = (i_1, i_2, ...i_l)$ be an index vector for a loop nest of depth l . The affine computation decomposition of the loop nest onto a n dimension processor space is an affine function $\vec{C}$:

$I -> P, \vec{c}(\vec{i}) = C\vec{i} + \vec{\gamma}$, C is a $n*l$ linear transformation matrix and $\vec{\gamma}$ is a constant vector.

Mathematically, the linear data and computation decomposition are represented by the matrices D and C from the above definitions.

In our model, all the statements within a loop nest are treated as a single unit. For iteration $\vec{i}$ of every loop nest, the affine computation decomposition function $\vec{c}(\vec{i})$ specifies the virtual processor which executes all statements of iteration $\vec{i}$. We do not consider finding separate affine function for each statement within the loop nest.

4. LINEAR DECOMPOSITION METHOD

4.1 The Basic Linear Decomposition Method

The method of data and computation decomposition without data reorganization is suitable for do-all loop. The relationship between data and computation is expressed by the following theorem: Let the computation decomposition for loop nest j be $\vec{c}_j$ and the data decomposition for array x be $\vec{d}_x$. Let $\vec{f}_{xj}$ be an array index function for array x in loop nest $_j$. For all iterations $\vec{i}$, the elements of the array will be local to the processor that references those elements if and only if

$D_x(f^k{}_{xj}(\vec{i})) + \vec{\delta}_x = C_j(\vec{i}) + \vec{\gamma}_j$ [7] (4.1.1)

Communications due to mismatches in the linear transformation part of an affine decomposition are more expensive than those due to mismatches in the offset part. We relax the equations to allow limited types of communication. We use the version of Eq. (4.1.1) that omits the offsets, Eq. (4.1.2). This will lead to a solution where the linear decompositions have no data-reorganization communication, but may still have nearest neighbor communication due to offsets. We refer to these decompositions as basic linear decompositions.

$D_x F^k{}_{xj}(\vec{i}) = C_j(\vec{i})$ (4.1.2)

We simplify Eq. (4.1.2), by eliminating the iteration space vector $\vec{i}$ on both sides of the communication equation to give $D_x F^k{}_{xj} = C_j$, there can be many possible solutions to this system of equations, including the trivial solution that assigns all the computation and data to a single processor. The aim, however, is to find a solution with the maximum degree of parallelism. Mathematically, this corresponds to finding linear data and computation decompositions such that for all loop nests j, rank of matrix C_j is as large as possible.

From above we can conclude: If all loop nests and array decomposition matrix C、D in a program are zero matrixes $C_j = \vec{0}, D_x = \vec{0}$, then there is no parallelism in this program. All loop nests and arrays are distributed to a single processor, and the program execute in serial. When matrixes C、D are full space, the parallelism is maximized.

4.2 Finding Offset Decomposition

In this section, we will discuss how to find the offset decomposition. From Eq. (4.1.1), Eq. (4.1.2) and

definition 1, we can calculate the computation displacement:

$\vec{\gamma}_j = D_x \vec{k}^k{}_{xj} + \vec{\delta}_x$ (4.2.1)

Then the array displacement can be calculated by the following equation:

$\vec{\delta}_y = \vec{\gamma}_j - D_y \vec{k}^k{}_{yj}$ （4.2.2）

We already know the data decomposition matrix and offset vector $\vec{k}$, then we can solve the above equation by a simple greedy strategy.

4.3 Improving with Read-Only Data Recognition

There are many programs we can't find the parallelism using this method. But we find that the program can be parallelized when we analyses it by hand. So what is the problem? After testing several examples, we find these examples have a same characteristic, consider the code in Fig. 2. There are two array access functions in loop nest 1 for array B, if we analyses the code using the method we described above, we can not find the solution for the equation derived from 3.1.2. So the code cannot be parallelism, but we can translate the program into parallel code if we analyses it by hand. How do we solving this problem? We solve this problem in three steps. Firstly, before we run the decomposition pass, we add a pre-pass processing the program, in the pre-pass, we add recognition for read-only data, after pre-pass, in output program of the decomposition pass, an annotation be added for the read-only data. Secondly, we do decomposition pass without considering the read-only array B in the loop nest 1, we will find a decomposition for array A and C. We solve the problem of array B in the next pass-code generation pass. In code generation pass we translate the output program of the decomposition pass into a SPMD program with message-passing. In this pass, the read-only arrays are sent to each processor with broadcast method. Other arrays receive and send are based on their array decompositions. Using this method, we can translate more serial programs into a parallel code. In the next section, we will analyses this example with the improving methods.

Recognition of read-only array is a common technique for improving the performance of parallel machines. The method we use in this section maintains the degree of parallelism inherent in the read-write data without introducing additional communication.

5. A SIMPLE EXAMPLE

```
Int A[N], B[N], C[N][N]

for i1=1 to N     /* doall */
  A[2*i1] = B[i1] + B[2*i1]

for i1 = 1 to N   /* doall */
  for i2= 2 to N do
    C[i1][i2] = C[i1][i2-1] + B[i2]
```

Figure 2

First, we analyses the program with original decomposition method. We find the array access matrix F, in the above example, the array access matrixes F in the first loop nest are: $F_{A1}{}^1 = [2\ \ 0]$, $F_{B1}{}^1 = [1\ \ 0]$, $F_{B1}{}^2 = [2\ \ 0]$; the array access matrixes F in the second loop nest are: $F_{C2}{}^1 = F_{C2}{}^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $F_{B2}{}^1 = [0\ \ 1]$

Based on Eq. (4.1.2), we can give the following equations:

$$D_A F_{A1}{}^1 = D_B F_{B1}{}^1 = D_B F_{B1}{}^1 = C_1$$
$$D_C F_{C2}{}^1 = D_C F_{C2}{}^2 = D_B F_{B2}{}^1 = C_2$$

There is no solution for these equations. So using original decomposition the program can only be executed in serial. Then we analyses the program with the improving decomposition method. Firstly, we run the pre-pass, array B is annotated with read-only. Secondly, we don't consider the decomposition for array B, we use original method to find the decomposition for array A and C. We can change the equations to the following forms:

$$D_A F_{A1}{}^1 = C_1$$
$$D_C F_{C2}{}^1 = D_C F_{C2}{}^2 = C_2$$

Then $C_1 = [2], C_2 = [1\ \ 0], D_A = [1]$, $D_C = [1\ \ 0]$.

Thirdly, in MPI code generation pass, we can produce the send and receive code based on the decomposition of array A and C, and we broadcast array B to each processor. Finally, we compile and run the output program with MPI compiler. The result of the program is correct.

6. CONCLUSIONS

Parallel compiler is an important part of the high-Performance parallel machines, and it's a key technology for realizing super computing. In this paper, we only present a linear decomposition method that automatically finding computation and data decomposition without data reorganization communication within a procedure, and we improved this method with read-only recognition. The method described in this paper has been implemented in our compiler system. This work has successfully integrated the SUIF [12] compiler with a Distributed Shared Memory (DSM) system. It forms a portable parallelization environment that can convert sequential FORTRAN or C programs to a parallel version that runs on a cluster of Symmetrical Multiprocessors (SMPs). This system can also serve as a test-bed for future work on the Automatic Parallelization Environment (APE). And we have applied the compiler to some benchmark programs-PPOPP and NPB. Through the tests, we verified the correctness of the method, but the effectiveness of the method need to be improved. Using this method, we can find a single data decomposition for each array throughout the entire program, but for most programs, we can not find a single data decomposition for each array throughout the entire program, so data reorganization is inevitable, finding data and computations decomposition that has data reorganization communication is a problem we will solve in the future work.

7. ACKNOWLEDGEMENTS

We would like to thank associate professor Zhang Ping, without her enthusiasm and support; the work would not be possible. Thanks also to Yuan Yao, XueRong Gong, YaNan Shen, HongTu Ma, Min Liu and ShanShan Wang. It is a great pleasure to work with them.

8. REFERENCES

[1] Y.L. Jin， Y.Q. Huang, et al, Trends and Key Technologies of High Performance Computers. Engineering Science, June 2001, vol.3 （6）:1-8.

[2] High Performance Fortran Forum. High Performance Fortran Language Specification, November 1992. Version 0.4.

[3] S. Hiranandani, K. Kennedy, et al, Evaluating Compiler Optimizations for Fortran D. Journal of Parallel and Distributed Computing, 1994, 27–45.

[4] B. Chapman, P. Mehrota, et al, Dynamic Data Distributions in Vienna Fortran. Proceedings of Supercomputing ’95, December 1995, 284–293.

[5] High Performance Fortran Forum, High Performance Fortran Language Specification, ver. 2, January 1997.

[6] J. Subhlok, B. Yang, A New Model for Integrated Nested Task and Data Parallel Programming. Proceedings of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, June 1997.

[7] M. Hall, J. Anderson, et al, Maximizing Multiprocessor Performance with the SUIF Compiler. IEEE Computer, December 1996, 29(12):84–89.

[8] C. Polychronopoulos, M. Girkar, et al, The Structure of Parafrase-2: An Advanced Parallelizing Compiler for C and Fortran. Workshop on Language and Compilers for ParallelComputing, August 1990.

[9] K.Kennedy, K.S.McKinley, Optimization for parallelism and data locality. In Proceedings of the 1992 ACM International Conference on Supercomputing, July 1992,323-334.

[10] D. Kulkarni, K. G. Kumar, et al, Loop partitioning for distributed memory multiprocessors as unimodular transformations. In Proceedings of the 1991 ACM International Conferenceon Supercomputing, June 1991,pages206-215.

[11]K.Kennedy, U.Kremer, Automatic Data layout for High Performance Fortran. Proc supercomputing 95[C], San Diego, calif, 1995

[12] Stanford Compiler Group, SUIF Compiler System Version1.0. US: Standford University,1994.

Grid Resource Scheduling Based on Negotiation among Agents

Shunli Ding, Qiang Chen, Jingbo Yuan
Department of Computer Engineering, Northeast University at Qinhuangdao, Qinhuangdao, Hebei, China
Email: dingsl@163.com, sdchenq@tom.com, jingboyuan@hotmail.com

ABSTRACT:

This work focuses mainly on Resource Scheduling in Grid Resource Management. The grid resources are abstracted highly to agents. This paper analyzes several common agent-models and put forward an improved multi-agent model for Resource Scheduling. During the procedure of scheduling, these agents will improve the ratio of success of Resource Allocation and Scheduling with the negotiation.

Keywords: Grid, Resource Scheduling, Agent, and Negotiation.

1. INTRODUCTION

Grid centralizes kinds of distributed resources and provides kinds of service, to supply a managing environment for synthesized problems [1]. Resource management can be departed to resource discovering, resource scheduling, mission referring and monitoring in a grid, therein, resource scheduling is the core problem. The intention of resource scheduling in Grid is to complete the submitted tasks and meet the demands of users. Resource scheduling matches all kinds of resources in Grid to find out the best and most reasonable manner and policy for resource distributing and scheduling.

The important problem which grid faces to is how to complete cooperated resource sharing in dynamic, heterogeneous and distributed virtual organizations and to settle one problem together. According to this, this paper designed an Agent-Negotiation Model to schedule resources more effectively.

2. RELATED WORKS

In multi-agent systems, the agents complete the negotiation with the method of information interacting. This procedure has mainly two kinds of information, preference (service describing), which is given out by the requestor, and ability (service capacity), which the provider gives out. This passage, we'll introduce a common useful broker model.
Broker Model: it receives the advertisement information that the requestor and provider give out. The broker can understand the preference and the capability and plays a router between them.

We analysis the Broker Model with the theory of queuing and get a conclusion: the Broker plays a role of the router between the requestor and the provider, so it can make the workload of the system more parallel. But, because of the structure of the broker (star structure), which is the model's inherent default, the throughput of the system will be severely affected when the number of agents increases.
Coordinating to settle one problem with the negotiation of agents is the main aim of multi-agent systems. Researchers advanced some strategies and methods to build multi-agent systems. The literature [2] adopted an algorithm of parameter engineering and feeling on scheduling jobs. The literature [3], aiming at the local area Grid environment, designed a local area Grid scheduler based on application level performance prediction. The literature [4] advanced a multistage multi-issue negotiation framework based on intelligent agents. The framework advanced an improved negotiation agenda through dividing the received negotiation issue. The literature [5] studied the negotiation algorithm, improved the multi-criterion negotiation and added the mechanism of agents allying. The literature [6] advanced a multi-stage negotiation model oriented virtual warehouses and analyzed the negotiation procedure among multi-agents. But, none of the above researches expressed how to make the workload of systems more reasonable, that is, how to balance workload.

The most different point of this paper compared with the above-related work is: it considered the workload balance problem in the system on building the multi-agent system. According to the work theory and negotiation mechanisms of the basic broker model, the paper designed a new Multi Agent-Negotiation Model.

3. THE AGENT-NEGOTIATION MODEL

The basic broker model is easy to bring forth bottleneck. One important reason is that there are too many communications between the broker agent and provider agents, and the broker is also in charge of transmitting the final result to the requestor, while the jobs of requestor and provider agents are so simple that the workload of the system is extreme imbalance. The Agent-Negotiation Model in this paper improves these disadvantages and will be discussed in detail in the following paragraphs.

3.1 Agent Creation

The agent is the composing element of the system of the Agent-Negotiation Model, and it is created by highly abstracting the resource information and requestors. The method is to quantify the web address, geography position, workload information and so on, which abstract the specific resources and requestors to agents with different parameters.

3.2 Multi Agent-Negotiation Model

For improving the basic broker agent model, this paper designed a new multi Agent-Negotiation model.

The whole system is composed of two parts: the bottom layer and the top layer. The bottom layer includes only node agents that transmit supplying and demanding information. The top layer includes three agents with specific jobs. The Communication agent is in charge of classifying the received information according a certain rule and transfers the information to the Broker; the Broker registers every agent by maintaining two-dimension tables; the Negotiation agent receives the related parameters of agents sent by the broker, and, by calling the matching algorithm, it selects an

agents' group that takes part in this negotiation and then sends the negotiation command to bottom agents.

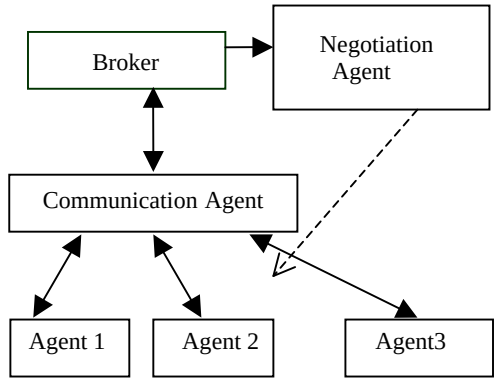


Fig. 1. Multi Agent-Negotiation Model

The ultimate innovation of the agent-negotiation model advanced is: The negotiation agents vote in several agents that participate in one negotiation process, besides it specifies a C-agent to take charge information alternation and coordination work. Because the C-agent belongs to the bottom layer and it is not decided in every negotiation procedure, we use a dash line to express the relationship. The advantage of this improvement is: at one time, there will be more than one group of negotiation agents exists in the system, which makes the system deal with the requestors of consumers in parallel manner. The bottom agents take over part jobs of negotiation and relieve the quantity of communication between negotiation agent and bottom agents, so the overload of the system is more balanced.

In the following paragraph, we will define the negotiation model and illustrate the negotiation process in detail.

3.3 Related Definition of Negotiation Model

According to different methods, the negotiation among agents is classified into dynamic, static, concentrated, distributed and so on .In the paper we adopt dynamic and concentrated model structure. The structure needs a Center Arbitration, which the broker in Fig. 1. severs as that one. The Center Arbitration Responses for managing global ability lib, receiving information distributed by bottom agents and transmitting the agent information satisfying the specific requestor negotiation condition to the negotiation agent.

This paper has abstracted resources and requestors to agents with parameters. The definition of these parameters is the basis of the negotiation process.

Advertisement Information ***ad***: The bottom agents send some messages to the communication agent with ability of a kind of service. Record to make: ***ad(c, a***), ***c*** stands for the class of some resource and *a* stands for the agent, which sent the message.

Agent Registering Information ***reg***: the communication agent sends this to the broker. When the communication agent receives the ***ad*** message, it will classify them according to the resource ability or the request and register them to the broker. Record to make: ***reg***(***c***, ***a, port, f***), ***port*** is the Port Number of the agent; ***f*** is the mark of one class.

Capability Table ***Ctable***: The broker put the resource to the table of capability, according the received ***reg***.

The Request Queue ***Rqueque***: According to the order that the request comes, the broker numbers them to a queue. The method of get () takes out the request according the order that is FIFS.

The Matching Set *A*: If a simple agent can meet one request, the broker will optimize the Matching Set; select the closest agent, Min (A), which is the Matching of minimum, decreasing the resource wasting.

The Negotiation Agent Group ***Group***: If one service needs more than one agent, the broker will send a group which can meet the negotiation request to the negotiation agent, ***Group*** (***agents***, ***port***), ***port*** is the port number of the requestor, ***agents*** is the set of agents that meet the request.

According to the maximum and minimum theory in the field of Mathematics, we select some agents satisfying the requestor from the agent set transmitted from Group, and get a local optimum answer. We don't want to adopt the global optimum answer because it need so big amount of calculation and time that it will affect the real-time performance of the system.

3.4 Negotiation Procedure

The negotiation begins from the broker. At the beginning, the broker selects one from the requestor queue, and, which one will be selected is a mission scheduling problem, which is not the issue of this paper so we use a simple method, FIFS (First in First Service).

The concrete negotiation steps:

1) The requestor advances some request to an agent a[i];
2) a[i] turns the request to a special capability c[k], if c[k] can't meet the requestor, jumping to (4) ;
3) a[i] selects a provider set p[r] which can satisfy c[k] using its derivation rule and examining model. If p[r] is not empty, it means there is spare provider; if c[k]=NULL, it will return to requestor and the negotiation will end;
   For (j=1 to |p[i]|)
   If S [i, j, k] =Idle then p[r] =p[r] Y {p[j]};
   If p[r]! =NULL then return p[r];
4) a[i] can't supply the service and send a request of negotiation to the broker;
5) When the broker receives the negotiation requestor, it will find the agent set A[r] which can supply the service, meanwhile, it will select the C-Agent;
   A[r] =NULL
   For (i=1 to |A|)
   If S [i, k] =Idle then A[r]=A[r]Y{a[i]};
6) If the A[r]!=NULL, the C-Agent will send negotiation command to every agent in set A[r] and waits for replying from each one; if A[r]=NULL, the C-Agent will send a failure message to the requestor;
7) One agent who receives the negotiation requestor will do like the second step, examine his state and return the result to C-Agent;
8) After receiving the reply from the agents, the C-Agent will check whether every agent is busy. If it is true, it will return failure message; otherwise, return the provider set;
9) The negotiation requestor a[i] receives the message from the C-Agent. According to the message, it will return failure message or the final result to the requestor. Finally, this negotiation procedure ends up.

4. PERFORMANCE TEST AND RESULTS

We use excellent simulation software on grid agent—Java Agent Development Framework (JADE) that simulates the Agent-Negotiation model. The experiment will be divided into two groups: (1) the traditional broker model; (2) the Agent-Negotiation model advanced in this paper.

4.1 Algorithm

At first, to take some initialization operation for the system, that is namely determining system size. For instance, before each simulation experiment, we set the request agents' number of the system as 100, expressed by variable *r* and the provider agents' number as 200 expressed by variable *p*. The broker will classify and store the advertisement information sent by the provider. This paper limits the information sent by the provider, namely only four kinds of resource information: Memory, Hard Disk, Web, and CPU, corresponding classification parameters respectively 1, 2, 3 and 4.

Variable *count* records every successful "Resource Scheduling", thus the success ratio of the system at a certain scale is (*count*/*r*)*100%.

4.2 Test Procedure

In general, we will conduct experiment twice; each experiment has different initialization parameters (two models have the same parameters in one experiment). Meanwhile, to enhance universality, each experiment conducts six emulations and it gets the average value of success ratio.

System initialization parameters in first experiment: *r*=500, *p*=1000

Value of the *count* in the first group:
282，301，294，295，290，291
The average of success ratio: 58.44%.

Value of *count* in the second group
403，408，421，385，391，423
The average of success ratio: 81.03%

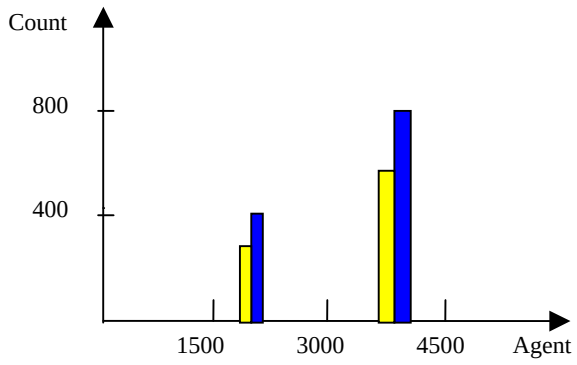


Fig. 2. Experiment Results

System initialization parameters in second experiment: *r*=1000, ***p***=2000
The average of success ratio in the first group: 60.32%;
The average of success ratio in the second group: 81.37%;

Fig. 2. denotes experiment results. Abscissa represents the agent total (*r*+*p*) in the system, and ordinate represents corresponding success count. Yellow brand represents the first group data, and, the blue brand represents the second group data.

4.3 Results

Through the comparison between the two experiments in the above section, we can make out clearly that the success ratio in the second group is nearly higher than that in the first group by 25%. Therefore, by validation of the simulation experiment, we can draw a conclusion that the new Agent-Negotiation model brought up by this paper can provider much better service quality.

5. CONCLUTIONS AND FUTURE WORK

The fundamental function of the grid resource scheduling system is to accept the resource request, which comes from computers in the grid environment, to assign specific resource to the requestor, and to schedule advisably corresponding resources to keep resource requestors carry out frequently. The Multi-Agent Model advanced in this paper, through negotiations among agents, can effectively satisfy tasks' request to resources and improve practical throughput of the system. Simultaneously, the bottom agents undertake massive negotiation tasks, which more balance load of the whole system.

Next emphasis of the problem is improving processing capability of negotiation agents. This is not only structural kernel, but also the key point of raising system efficiency. We set different negotiation strategies for the negotiation agent, to enable it to select optimum program based on requestors and present resources, and to find out more proper C-Agent to be responsible for detail coordinating tasks according to the overload of bottom agents.

6. REFERENCES

[1] I. Foster, C. Kesselman, "The Grid: Blueprint for a new computing Infrastructure", 1998.

[2] D. Abramson, R. Buuya, et al, "A computationa1 economy for grid computing and its implementation in the nimrod resource brokered", Future Generation Computer Systems, 2002, 18(8): 1061-1074.

[3] J. Frey, T. Tannenbaum, et a1, "Condor-G: a computation management agent for multi-institutional grids", Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing, Britain, IEEE, 2001, 55-63.

[4] L.M. Wang, H.K. Hong, "A Multistage-Based Framework for Multi-Agent Multi-Issue Negotiation", Journal of Computer Research and Development, 2005, 24(11): 1849-1855.

[5] D.P. Cai, S.J. Zhang, "Construction of Multi-Agent Arrange-alliance and Its Arithmetic Analysis in DDSS", Computer Engineering, Dec. 2005, 31(23).

[6] R. Fan, Q.Z. Zhou, "Multi-Agent based Alliance Corporation in Military Oil Virtual Warehouse", Computer Application, 2005, No.12, 33-35.

Automatic Computation and Data Decomposition Algorithm with No Communication

Han Lin, Zhao Rongcai, Pang Jianmin, Dong Chunli
National Digital Switching System Engineering &Technological R&D Center
Zhengzhou, Henan China
Email: stroller_hl@163.com

ABSTRACT

For distributed memory machines the speed that processor accesses local memories is much faster than its speed to access remote memories. Thus how to decompose data and computation properly to achieve maximum parallelism and minimum communication is a key issue of automatic parallel compilation. In this paper, the authors present an automatic decomposition algorithm based on constraint equations. Using the algorithm, a data and computation decomposition result with no communication can be achieved. By releasing these constraint equations, the authors get decomposition result with more parallelism. As the result may have communications, so an improved method to eliminate some communications by data replication is also presented.

Keywords: parallel recognition compiler, computation partition, data distribution.

1. INTRODUCTION

On large-scale parallel machines, each computation node has its own memory， and the communications between nodes are achieved by message mechanism. As the speed of processor's accessing local memory is much faster than the speed of its accessing remote memory, then how to analyze serial codes in the process of automatic parallelism properly, find the parallel codes and decompose the computation and data reasonably to achieve local data reference have become the key points of automatic parallel compilation.

We call mapping computation to each computer node's local memory "computation partition" and name mapping data referenced by computation to that processor's local memory "data distribution". *Anderson* and *Lam* contribute a great deal of work to computation partition and data distribution. They put forward an algorithm which can automatically decompose computation and data[1]. This algorithm, which can be denoted by linear loop index, is suitable for the visit of array elements referenced by loops, that is, the subscript of array is affine function of loop index, and the algorithm presents decomposition result by analyzing constraints in parallel computation and data. We find that the algorithm can't satisfy our decomposition requirements and may bring in a mass of communications in certain conditions, so we make an improvement of it and get some favorable test results.

The algorithm discussed in the paper is adapted to both distributed and shared memory machines. We try to find a static decomposition for both loop nest and array, and present it as matrix or inequation. In the process of the algorithm analysis what we consider is mapping computation and data to virtual processor， that is, we suppose virtual processor's number and dimension satisfy decomposition requirement without mapping between virtual processor and physical processor.

2. SUMMARY OF COMPUTATION AND DATA DECOMPISITION

The technology of seeking maximum parallelism and locality within a loop nest has been mature [2,3], and many researchers have also applied themselves to how to map a given loop nest to scalable machines effectively [4,5,6]. We call optimization for a loop nest local analysis. How to make an optimization analysis to the full scale, that is, how to make an analysis of maximum parallelism and locality to multiple loop nests, is a problem to be solved.

It is difficult to get an analysis result. *kennedy* and *kremer* made it clear that the problem of finding the best data distribution is a NP complete problem [7], the reasons are that, firstly, there are many choices to make computation and data analysis, then how to choose the optimization is the problem we are facing; secondly, the decomposition should be taken into consideration in a whole program, for the decomposition of data and computation is connected with each other and infects each other.

We suppose that every loop nest in the program has been optimized and has found maximum parallelism of outmost loops by unimodule transformation technique [3,8]. Thus every loop nest transforms to a fully-permutable loop nest. The algorithm analyzes multiple loop nests in a program and finds out a set of constraint equations to give a computation and data decomposition result by solving these equations.

Define loop nest space L for a l-dimension space, used to express a loop nest with l depth, and define that any point of this space stands for an iteration of this loop nest, signed by $\vec{i} = (i_1, i_2, ...i_l)$;define array index space A for a m-dimension space, used to express a m dimension array, and define that any point of this space stands for an element of this array, signed by $\vec{a} = (a_1, a_2, ...a_m)$; processor space P for a n-dimension space, used to express a n-dimension processor, and any point of this space stands for the location of this processor, signed by $\vec{p} = (p_1, p_2, ...p_n)$.

Define array access affine function $f(\vec{i}) = F\vec{i} + \vec{\beta}$ for a map from l-dimension loop nest space to m-dimension array subscription index space, where F is a $l * m$ linear transform matrix and $\vec{\beta}$ is a constant vector; array distribution affine function $d(\vec{a}) = D\vec{a} + \vec{\delta}$ for a map from m-dimension array subscription index space to

n-dimension virtual processor space, where D is a $n*m$ linear transform matrix and $\vec{\delta}$ is a constant vector; computation partition affine function $c(\vec{i}) = C\vec{i} + \vec{\gamma}$ for a map from l-dimension loop nest space to n-dimension virtual processor space, where C is a $n*l$ linear transform matrix and $\vec{\gamma}$ is a constant vector. The goal of our algorithm is to seek a set of decomposition affine functions for every loop nest and every array to indicate the mapping from data and computations to virtual processors.

3. DESCRIPTION OF ALGORITHM

Suppose that a loop nest is with l -depths and the computation partition matrix is C , which means the parallelism of this loop nest is $rank(C)$, and $rank(C) = l - \dim(\mathrm{N}(C))$. The nullspace of partition matrix is the computation needed to map to the same processor in physical meaning. So maximum parallelism means finding a linear decomposition in which $rank(C)$ is as large as possible, that is, the dimension of nullspace $\mathrm{N}(C)$ is as small as possible. The introduction of nullspace makes us express the data and computation needed to map to the same processor in a concise way. When a new decomposition constraint is added, we only need to renovate nullspace and needn't to calculate decomposition equation again. So in the final realization of our algorithm, we can first get the nullspace of decomposition matrix, then the corresponding decomposition matrix.

Data and computation is affected with each other during decomposition. We use interference graph $G_s(V_c, V_d, E)$ to describe the array referenced information in loop nests. In the graph vertices V_c stands for a loop nest, V_d for array, if an array is referenced by a loop nest, then an undirected edge E exists between them.

The computation mapping to certain processor and the data it referenced should be distributed to local processor relatively considering the relationship between computation and data. Suppose the computation partition matrix of loop nest j is C_j, the data distribution matrix of array X is D_x, and the κ th array access function to array X in loop nest j is function ${f^k}_{xj}$, then all the iteration $\vec{i}$ in this loop nest is applicable to the formula:

$$D_x({f^k}_{xj}(\vec{i})) + \vec{\delta}_x = C_j(\vec{i}) + \vec{\gamma}_j \text{ [1]} \quad (3.1)$$

The decomposition which accords with the above formula completely is no communication decomposition. vector $\vec{\delta}$ and vector $\vec{\gamma}$ in the formula points out the necessary offset in decomposition, which we will ignore in the realization of algorithm till we find out computation partition matrix C_j and data distribution matrix D_x. Thus the formula above can be reduced by:

$$D_x {F^k}_{xj}(\vec{i}) = C_j(\vec{i}) \quad (3.2)$$

Linear decomposition algorithm limits the constraint. The computation and data should abide by decomposition through studying the conditions the decomposition matrix and the nullspace need to satisfy when the above formula is tenable. We find that the algorithm can result in much communication by study and test when the code realizes. We shall list the constraints in *Anderson* and *Lam's* papers and the constraints we have found in this paper, then we can get a set of no communication decomposition based on these constraints, the following constraint equations are based on formula 3.2.

Synchronization constraint equation: This equation generates synchronization constraint on the nullspace of the linear computation decomposition matrix. Consider synchronization constraint existing in a loop nest, the outmost loops of this loop nest have parallelism, while the inner loops should be mapped to the same processor.

Suppose the depth of a loop nest j is l, the outer loops $1 \cdots s$ can parallel, and inner loops $q = (s+1) \cdots l$ existing synchronization, then iteration $\vec{i}$ and $\vec{i} + \vec{e}_q$ needed to be partitioned to the same processor, where $\vec{e}_q$ is the qth element vector of l-dimension space. Formula description as $C_j(\vec{i} + \vec{e}_q) = C_j(\vec{i}) \Rightarrow C_j((\vec{i} - \vec{e}_q) - \vec{i}) = \vec{0} \Rightarrow C_j(\vec{e}_q) = \vec{0}$, that is, $\vec{e}_q \in \mathrm{N}(C_j)$ (3.3)

Iteration constraint equation: This equation guarantees that when multiple iterations of a loop nest access a certain element of an array, this element should be partitioned to the same processor.

If two iterations $\vec{i}_1$ and $\vec{i}_2$ of loop nest j access the same element of array X when ${F^k}_{xj}(\vec{i}_1) = {F^k}_{xj}(\vec{i}_2)$, that is, ${F^k}_{xj}(\vec{i}_1 - \vec{i}_2) = \vec{0}$. Letting $\vec{t} = \vec{i}_1 - \vec{i}_2$, $\vec{t} \in \mathrm{N}(F^k{}_{xj})$. Since $\mathrm{N}(D_x F^k{}_{xj}) \supseteq \mathrm{N}(F^k{}_{xj})$, then $\vec{t} \in \mathrm{N}(F^k{}_{xj})$ implies that $\vec{t} \in \mathrm{N}(C_j)$. This leads to the following constraint:

$$\forall \vec{t} \in \mathrm{N}(F^k{}_{xj}), \vec{t} \in \mathrm{N}(C_j) \quad (3.4)$$

Single cycle constraint equation: This equation depicts the constraint of the array distribution when a loop nest pays many visits to a certain array.

Suppose ${F^1}_{xj}$ and ${F^2}_{xj}$ denote two access affine function of loop nest j to array X , and then in interference graph, a simple cycle exists between loop nest j and array X . From formula 3.2, equation $D_x {F^1}_{xj}(\vec{i}) = C_j(\vec{i})$ and $D_x {F^2}_{xj}(\vec{i}) = C_j(\vec{i})$ are exist, this leads to $D_x({F^1}_{xj}(\vec{i}) - {F^2}_{xj}(\vec{i})) = \vec{0}$, that is, $range({F^1}_{xj}(\vec{i}) - {F^2}_{xj}(\vec{i})) \in \mathrm{N}(D_x)$ (3.5)

Multiple cycles' constraint equation: This equation ensures the constraint when an array is referenced by multiple loop nests. When there are multiple paths between two vertices in interference graph, then it is possible for the loop nests to cause conflicting requirements on the decomposition of the arrays. We should also take multiple cycles caused by access propagation into consider.

Suppose vertices $V_x, V_y \in (V_d \cup V_c)$ in graph is $G_s(V_c, V_d, E)$, if there is a cycle in the graph $(V_x, \cdots, V_y, \cdots, V_x)$, then there are multiple distinct paths from V_x to V_y . From formula 3.2, equation $D_x {F^k}_{xm}(\vec{i}_m) = D_y {F^k}_{ym}(\vec{i}_m) = C_m(\vec{i}_m)$ and

$D_x F^k{}_{xn}(\vec{i}_n) = D_y F^k{}_{yn}(\vec{i}_n) = C_n(\vec{i}_n)$ exist, this gives the following equation: $D_y = D_x F^k{}_{xm}(\vec{i}_m) F^k{}_{ym}{}^{-1}(\vec{i}_m)$ and $D_y = D_x F^k{}_{xn}(\vec{i}_n) F^k{}_{yn}{}^{-1}(\vec{i}_n)$, thus following constraint:

$$range\left(F^k{}_{xm}(\vec{i}_m) F^k{}_{ym}{}^{-1}(\vec{i}_m) - F^k{}_{xn}(\vec{i}_n) F^k{}_{yn}{}^{-1}(\vec{i}_n)\right) \in \mathrm{N}(D_x) \quad (3.6)$$

Computation propagation constraint equation: This equation guarantees computation constraint is propagated to data constraint using formula 3.2.

Suppose two iterations $\vec{i}_1$ and $\vec{i}_2$ in loop nest j are mapped to the same processor, then the data of array X they access should be mapped to the same processor. We have $C_j(\vec{i}_1) = C_j(\vec{i}_2)$, thus for $\vec{t} = \vec{i}_1 - \vec{i}_2$, from formula 3.2 $D_x F^k{}_{xj} \vec{t} = C_j \vec{t} = \vec{0}$, formally:

$$span\{\vec{s} \mid \vec{s} = F^k{}_{xj}\vec{t}, \vec{t} \in \mathrm{N}(C_j)\} \subseteq \mathrm{N}(D_x) \quad (3.7)$$

Data propagation constraint equation: This equation guarantees data constraint is propagated to computation constraint using formula 3.2.

Similarly, two iteration $\vec{i}_1$ and $\vec{i}_2$ in loop nest j should be mapped to the same processor if the data of array X they access mapped to the same processor.

Again, from formula 3.2 $D_x F^k{}_{xj} \vec{t} = C_j \vec{t} = \vec{0}$, thus $\vec{t} \in \mathrm{N}(C_j)$ when $F^k{}_{xj}\vec{t} \in \mathrm{N}(D_x)$, formally:

$$span\{\vec{t} \mid F^k{}_{xj}\vec{t} \in (\mathrm{N}(D_x) \cap range(F^k{}_{xj}))\} \subseteq \mathrm{N}(C_j) \quad (3.8)$$

Algorithm description: Before our decomposition pass, the parallelization analysis is raised, which tries to transform the input serial code using unimodular transformations to expose the maximum degree of loop-level parallelism, while minimizing the frequency of synchronization. Then decomposition algorithm takes as input the loop nests in the canonical form of nests of fully permutable loop nests, it analyzes the array accesses within the loop nest to calculate the mappings of data and computation onto virtual processors. It first builds interference graph $G_s(V_c, V_d, E)$ by analyzing relation between loop nests and arrays, then uses constraint equations 3.3~3.8 to get a set of nullspace of decomposition matrix. For each loop nest and for each array accessed in each loop nest, the algorithm outputs a system of decomposition matrix that describes the mapping from data and computation to virtual processors.

Final decomposition results contain decomposition matrixes and offsets. Using computation partition matrix C and data distribution matrix D with formula 3.1, we can calculate offset by the following equations.

Computation offset: $\vec{\gamma}_j = D_x \vec{\beta}^k{}_{xj} + \vec{\delta}_x$ (3.9)

Data offset: $\vec{\delta}_y = \vec{\gamma}_j - D_y \vec{\beta}^k{}_{yj}$ (3.10)

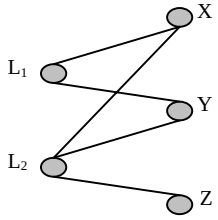
4. AN EXAMPLE

Consider the following code, which contains two loop nests L_1, L_2 and three arrays X, Y, Z. After parallelism analysis, we find all loops in these two loop nests have 'doall' parallelism.

(1) *for* $i:0$ *to* N *do*
  for $j:0$ *to* N *do*
    $Y[i, j] \mathrel{+}= X[i, j];$

(2) *for* $i:0$ *to* N *do*
  for $j:0$ *to* N *do*
    $X[i, j] = Z[i, j-1] + Y[i, j];$

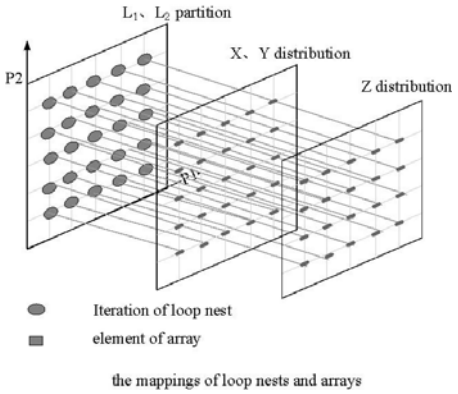
Corresponding interference graph $G_s(V_c, V_d, E)$:



Use decomposition algorithm to analyze this code, the final decomposition result expressed by matrix is:

$$C_{L1} = C_{L2} = D_x = D_y = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ and } D_z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The decomposition result shows that loop nest L_1, L_2 and array X, Y, Z can parallel in a 2-dimension space. Mapping loop nests and arrays to a 2-dimension virtual processor is as the following:



the mappings of loop nests and arrays

The figure shows computation and data are fully distributed in this 2-dimension processor with no communication.

5. ELIMINATE COMMUNICATION BY OPTIMIZATION

Replication of read-only data is a commonly-used technique for improving the performance of parallel machines. Now we present an example *sor.f* in benchmark *ppopp* for discussing the application of read-only data replication. Using our algorithm, the decomposition matrix is NULL because we have considered the question that multiple referenced of an array may not give a uniform decomposition, we release single cycle constraint equation to get another decomposition.

```
      Do   11  I = 1, 512
        Do   10  J = 1, 512
             a(J, I) = b(J, I)
10      continue
11    continue
      Do   21  I = 2, 511
        Do   20  J = 2, 511
             b(J, I) = a(J, I) + a(J − 1, I) + a(J + 1, I) +
     &       a(J, I − 1) + a(J, I + 1)
20      continue
21    continue
```

The result of decomposition is:

$$C_{L1} = C_{L2} = D_a = D_b = \left(\begin{matrix}1 & 0\\ 0 & 1\end{matrix}\right) + \left(\begin{matrix}0\\ 0\end{matrix}\right).$$

For array a is read-only and the reference of array a is not all local, so we replicate array a with bounds 1,. The array distribution is as the following, where p means a virtual processor:

$a_{30}\mathbf{a_{40}}a_{50}a_{41}$ P(4,0)	$a_{40}\mathbf{a_{41}}a_{31}a_{51}a_{42}$ P(4,1)	$a_{32}\mathbf{a_{42}}a_{41}a_{52}a_{43}$ P(4,2)	$a_{42}\mathbf{a_{43}}a_{33}a_{53}a_{44}$ P(4,3)	$a_{43}\mathbf{a_{44}}a_{34}a_{54}a_{45}$ P(4,4)
$a_{20}\mathbf{a_{30}}a_{40}a_{31}$ P(3,0)	$a_{21}\mathbf{a_{31}}a_{30}a_{41}a_{32}$ P(3,1)	$a_{31}\mathbf{a_{32}}a_{22}a_{42}a_{33}$ P(3,2)	$a_{32}\mathbf{a_{33}}a_{23}a_{43}a_{34}$ P(3,3)	$a_{33}\mathbf{a_{34}}a_{24}a_{44}a_{35}$ P(3,4)
$a_{10}\mathbf{a_{20}}a_{30}a_{21}$ P(2,0)	$a_{20}\mathbf{a_{21}}a_{11}a_{31}a_{22}$ P(2,1)	$a_{21}\mathbf{a_{22}}a_{12}a_{32}a_{23}$ P(2,2)	$a_{22}\mathbf{a_{23}}a_{13}a_{33}a_{24}$ P(2,3)	$a_{23}\mathbf{a_{24}}a_{14}a_{34}a_{25}$ P(2,4)
$a_{00}\mathbf{a_{10}}a_{20}a_{11}$ P(1,0)	$a_{10}\mathbf{a_{11}}a_{01}a_{21}a_{12}$ P(1,1)	$a_{11}\mathbf{a_{12}}a_{02}a_{22}a_{13}$ P(1,2)	$a_{12}\mathbf{a_{13}}a_{03}a_{23}a_{14}$ P(1,3)	$a_{13}\mathbf{a_{14}}a_{04}a_{24}a_{15}$ P(1,4)
$a_{01}\mathbf{a_{00}}a_{10}$ P(0,0)	$a_{00}\mathbf{a_{01}}a_{11}a_{02}$ P(0,1)	$a_{01}\mathbf{a_{02}}a_{12}a_{03}$ P(0,2)	$a_{02}\mathbf{a_{03}}a_{13}a_{04}$ P(0,3)	$a_{03}\mathbf{a_{04}}a_{14}a_{05}$ P(0,4)

By replicating array a with bounds 1, we eliminate the communication. This method is efficient for read-only array.

6. SUMMARY

This paper mainly describes an automatic computation and data decomposition algorithm with no communication based on constraint equations. We present all these constraints in the paper and make a formalized description through deep analysis of serial codes. The algorithm has been used in our parallel recognition compiler.

7. REFERENCE

[1] J.M.Anderson, M.S.Lam, Global optimizations for parallelism and locality on scalable parallel machines. In Proceedings of the SIGPLAN '93 Conference on Programming Language Design and Implementation, Albuquerque,NM, June 1993， 112–125.

[2] K.Knobe, J.D.Lukas, et al, Data optimization: Allocation of arrays to reduce communication on SIMD machies.Journal of Parallel and Distributed Computing, 1990，8:102-118.

[3] M.S.Lam, M.E.Wolf, Compilation techniques to achieve parallelism and locality. In Proceedings of the DARPA Software Technology Conference, April 1992， 150-158.

[4] C.H.Huang, P.Sadayappan, Communication-free hyperplane positioning of nested loops. In U. Banerjee, D. Gelernter, et al, editors, Languages and Compilers for Parallel Computing, Springer-Verlag, Berlin, Germany, 1992，186–200.

[5] D.Kulkarni, K.G.Kumar, et al, Loop partitioning for distributed memory multiprocessors as unimodular transformations. In Proceedings of the 1991 ACM International Conference on Supercomputing, June 1991， 206–215.

[6] K.G.Kumar, D.Kulkarni, et al, Deriving good transformations for mapping nested loops on hierarchical parallel machines in polynomial time. In Proceedings of the 1992 ACM International Conference on Supercomputing, July 1992， 82–91.

[7] K.Kennedy, U.Kremer, Automatic Data Layout for High Performance Fortran [A]. Proc Supercomputer[C]. San Diego, Calif, 1995.

[8] M.E.Wolf. Improving Locality and Parallelism in Nested Loops. PhD thesis, Stanford University, August 1992. Published as CSL-TR-92-538.

A New Approach to Load Balancing Algorithm in LVS Cluster System

Wei Ma
Center of Computing and Experimenting, South-Central University for Nationalities
Wuhan, Hubei 430074, P.R.China
Email: mw0626@163.com
Shijue Zheng, Li Gao, Jianhua Du
Department of Computer Science, Hua Zhong Normal University
Wuhan, Hubei 430079, P.R.China
Email: zhengsj@mail.ccnu.edu.cn

ABSTRACT

In order to raise the availability of single server and avoid overload, the employment of sever cluster is an effective solution. As a means of implementation technology for the server cluster, Linux Virtual Server (LVS) cluster technology employs a load balancer to dispatch requests to the back end cluster servers.
In a LVS system, how to provide services for more clients and reduce the average response time of requests is a critical topic. Pointing to the weakness of current algorithms used in LVS cluster, we present a dynamic load balancing algorithm which can improve the performance of the system. Through the experiment, we found that our algorithm has better load balancing and faster response time of requests.

Keywords: LVS, Load Balancing, Dynamic Algorithm.

1. INTRODUCTION

With the rapid development of Internet and explosive increase of web application, the load of web server grows exponentially, which requires better performance for the server. Some hot websites are faced with such problems as server overload and lengthy response time of the request. In order to raise the availability of single server and avoid overload, the employment of sever cluster is an effective solution. Server cluster arises as a strongly vital technology with the growth of Internet and web, and the realization of load balancing by means of request dispatching is an important method to elevate the system operation efficiency. As a means of implementation technology for the server cluster, Linux Virtual Server (LVS) cluster technology employs a load balancer to dispatch request to the back end cluster server, consequently raising the utilization of system resources, decreasing response time and improving the performance of the whole system.

2. SYSTEM ARCHITECTURE

The cluster system is composed of the following modules: clustered servers, a storage system, a load balancer, a network and clients.
Fig.1 shows the architecture of the cluster system, which is based on a client-server model. According to the layout of whole system, dedicated servers consist of several high performance servers, which form a typical distributed subsystem. It is in charge of original distribution of data, initial load balance, security policy, global admission control policy, and overall availability, etc. Each server has its own CPU, memory and disk storage [1,2,3].

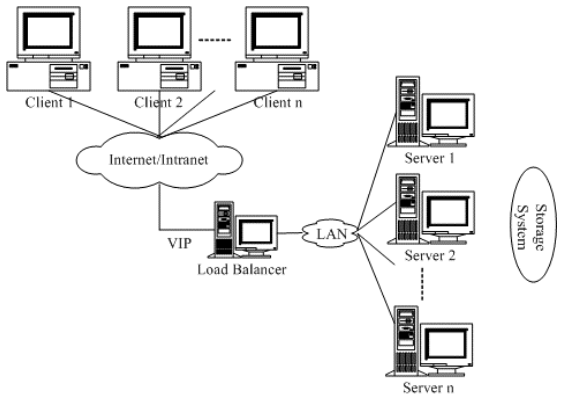


Fig. 1.Cluster System Architecture

Presently the big problem with load balancing technology in LVS cluster system is: In the process of making load balancing, server node can't dynamically and effectively give load feedback to the load balancer, thus unable to play its role very well in balancing. To solve this problem, we present a load balancing algorithm with dynamic load feedback mechanism pointing to the weakness of current algorithms used in LVS cluster.
The rest of the paper is organized as follows: In section 2 we describe the system architecture. In section 3 we address the current load balance technology and the algorithm. In section 4 we present the dynamic load-balancing algorithm. In section 5 we give laboratorial measurements for our algorithm, compare the performance with current LVS algorithm and provide the experimental results, which are based on LVS, cluster system. In section 6 we conclude our paper and discuss some future work related to our study.
The distributed architecture is based on a cluster of servers attending to a great number of client requests [3]. Clients are not expected to send requests directly to the servers. The requests go first to the balancer, which redirects them to the appropriate server taking into consideration issues such as block location and load balancing.
Cluster system has two important advantages compared with the single computer machine solutions. The first one refers to the ease that a server can be extended using more nodes. The cost of changes that must be performed is negligible, compared to the profit that the addition of new nodes introduces to the system. The second one refers to the requirement that, after some node's failure, the whole system should continue to operate correctly. The cluster system fully satisfies these two key factors keeping the overall cost low.

3. LOAD BALANCING TECHNOLOGY

Currently three technologies based on IP layer are usually used to achieve load balancing among servers. They are NAT (Network Address Translation), DR (Direct Route), and IP tunneling [4,5].

3 .1 Network Address Translation

In NAT mode, the service IP address will be configured on the load balance server. When a client sends request packets, the NAT server will convert the destination address to the address of the real server chosen and relay the packets to the real server, when the real server replies, the load balance server will change the source address, which is now the address of the real server to the service IP address and send the packets back.
The advantage of the virtual server via NAT is that real servers can run any operating system that supports TCP/IP protocol, real servers can use private Internet addresses, and only an IP address is needed for the load balancer.
The disadvantage is that the scalability of the load balance server via NAT is limited. The load balancer may be a bottleneck of the whole system when the number of server nodes increase to around 20 or more, because both the request packets and response packets are need to be rewritten by the load balancer.

3.2 Direct Routing

In this mode, all request and response packets need to pass through the load balancer, the load balancer in direct routing mode just schedules requests to the different real servers, and the real servers return replies directly to the users. Since in most applications, request will have far less bytes than response, so the load balancer can handle huge amount of requests, comparing to NAT. Thus using direct routing will greatly increase the maximum number of server nodes for a load balancer. The maximum throughput of the load balancer can reach over 1Gbps, even if the load balancer just has 100Mbps full-duplex network adapter.
The direct routing feature uses MAC-spoofing technology, so it requires that one of the load balancer's interfaces and the real servers' interfaces must be in the same IP network segment and physical segment as well.

3.3 IP Tunneling

Like in the load balancer via direct routing approach, it processes only the client-to-server half of a connection, and the response packets can follow separate network routes to the clients.
The disadvantage is that all servers must have "IP Tunneling"(IP Encapsulation) protocol enabled. Well the advantage is since servers connect each other using IP Tunnel, the load balancer and real servers can resides on different LAN, or even WAN.

3.4 Current Load Balancing Algorithms

According to different applications and network configuration, there are eight algorithms used to decide which real server will be used to the new coming connection.
RR: Round Robin, requests from the clients will be sent to the next available server in the cluster on a rotating basis. Jobs will be equally distributed among the available real servers.
WRR: Weighted Round Robin, assign jobs to real servers proportionally to real servers' weight. Servers with higher weights receive new jobs first and get more jobs than servers with lower weights. Servers with equal weights get an equal distribution of new jobs.
LC: Least-Connection, assign more jobs to real servers with fewer active jobs.
WLC: Weighted Least-Connection, assign more jobs to servers with fewer jobs and relative to the real servers' weight. This is the default.
LBLC: Locality-Based Least-Connection, assign jobs destined for the same IP address to the same server if the server is not overloaded and available; otherwise assign jobs to servers with fewer jobs, and keep it for future assignment.
LBLCR: Locality-Based Least-Connection with Replication, assign jobs destined for the same IP address to the least-connection node in the server set for the IP address. If all the node in the server set are over loaded, it picks up a node with fewer jobs in the cluster and adds it in the sever set for the target. If the server set has not been modified for the specified time, the most loaded node is removed from the server set, in order to avoid high degree of replication.
DH: Destination Hashing, assign jobs to servers through looking up a statically assigned hash table by their destination IP addresses.
SH: Source Hashing, assign jobs to servers through looking up a statically assigned hash table by their source IP addresses.
Algorithms mentioned above are suitable for different applications. But they are static loading balancing, which cannot feedback the real servers situation to the balancer. It is only decided by the balancer to distribute the requests to the real servers. Here we present a dynamic load balancing algorithm [6,7]. It is different from static algorithm and achieves better load balancing.

4. THE DYNAMIC ALGORITHM

In this section, we develop an analysis by combining a feedback mechanism with the weighted least connection algorithm. When clients send the requests to the cluster, the time and resource for processing the requests are different. They depend on many factors such as services types, current network bandwidth and the utilization of the real servers. So requests for different processing time will lead to load imbalance. Here we adopt a feedback mechanism algorithm, which can help balancer get the present load state of the real servers. The algorithm adjusts allocation proportions according to the real load state and capabilities of sever. A monitor program will be run on the balancer and gather global information of each server node.
In cluster applications, the most popular load-balancing algorithm is weighted least-connection which has better load balancing effect and distribute the requests to the server nodes with the least connections. But the number of connections can't reflect the load state of severs very well. Our algorithm is a dynamic feedback algorithm, which compute the load not only considering the connections but also depending on CPU utilization and response time. The basic idea of the algorithm is described as follow: Give a cluster with n server nodes Sn={S1, S2, S3,⋯Sn} ,C(Si) is the number of connections on node Si. U (Si) is the CPU utilization on node Si. R (Si) is the response time of the request which be sent from the load balancer. And i is the server node selected last time. The new request will be send to the server node m which matches the Eq. (1):

$$C(S_m)\times(U(S_m)+R(S_m))=\min\{C(S_i)\times(U(S_i)+R(S_i))\} \quad (1)$$

The algorithm adopt the dynamic feedback mechanism. And consider the connections, CPU utilization and response time (CCRT for short), which can reflect the server load situation. It adjusts the task allocation according to the feed back information. As a result avoid the overload on the server nodes and improve the system performance.

5. PERFORMANCE EVALUATION

In this section, we evaluate performance of the algorithm presented in Section IV. When the cluster begins to work, the balancer will send a special request to the servers periodically. Every server node will collect the CPU utilization and send it to the load balancer as a response. The load balancer computes the response time by the receiving time minus the sending time. And the new arrival request will be send to the server with the least load according to the equation mentioned in section IV.

We testify the algorithm based on LVS cluster system adopting Network Address Translation mode. And explore the load balancing effect by using Microsoft web application stress tool. It can simulate the web browser and send requests to the servers. We know the most important parameters, which can reflect the performance of the cluster is throughput and response time. In this experiment, the two parameters can be reflected by byte receive rate and TFB. We explore throughput and response time under different concurrent connections using the different algorithms. One is weighted least-connection algorithm. And the other is our dynamic one. The results are shown in Fig.1 and Fig.2.

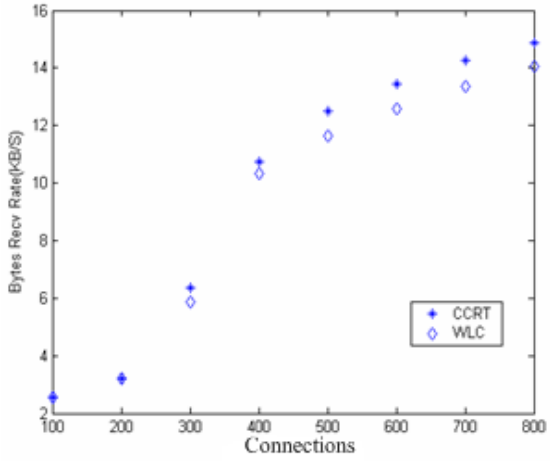


Fig. 1. Byte Receive Rate

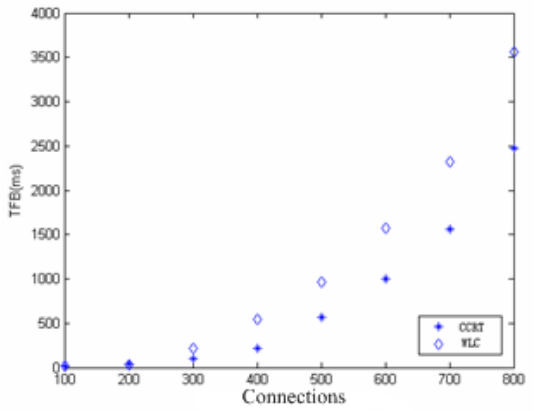


Fig. 2. Average TFB

6. CONCLUSIONS

In the paper, we propose dynamic load balancing in a cluster system to support more clients and reduce the average response time of requests. It consists of (1) cluster system Architecture, (2) load balancing technology, (3) analysis of the dynamic algorithm, and (4) performance evaluation, respectively. Through the experiment, we found that our algorithm has better load balancing, higher accepted rate and faster response time of requests than the weight least connection strategies.

7. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of P.R.China under Grant No. 60473085. We thanks Professor Liansheng Tan of the Department of Computer Science of Hua Zhong Normal University Technology for his supports of various aspects in setting up cluster system and thanks also go to all members of our research group especially Jianhua Du and Wanneng Shu for their efforts in the coding task.

8. REFERENCES

[1] S. J. Zheng, W. Ma, et al, “Analysis of Distributed Video-On-Demand Systerm Based on Cluster”，in Proceedings of 2004 International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES 2004), Hubei Science and Technology Press, Wuhan China, Sept. 13-16, 2004, pp.400~403

[2] Y.B.L.Jack, P.C.Wong, “Performance Analysis of Pull-Based Parallel Video Server”, IEEE Transactions on Parallel and Distributed Systems, VOL.11, NO.12, DECEMBER 2000.

[3] Aostolos Papagiannis, Dimitrios Lioupis, et al, “Design & Implementation of a Low-cost Clustered Video Server Using a Network of Personal Computers”, Proceedings of the IEEE Fourth International Symposium on Multimedia Software Engineering (MSE’02).

[4] Y. M. Teo, R. Ayani, “Comparison of Load Balancing Strategies on Cluster-based Web Servers”, Transactions of the Society for Modeling and Simulation, 2001

[5] H. Bryhni, E. Klovning, et al, “A Comparison of Load Balancing Techniques For Scalable Web Servers,” IEEE Network, Vol. 14, No. 4, pp. 58-64. Jul/Aug 2000.

[6] Y. F. Huang, C. C. Fang, “Load Balancing of Clusters VOD Servers”, Information Sciences 164(2004) 113-138

[7] A.M.I.Mohammed, X.D.Lu, “Performance of Dynamic Load Balancing Algorithm on Cluster of Workstations and PCs”, Proceedings of the Fifth International Conference on Algorithms and Architecture for Parallel Processing (ICA3PP’02)

Single Stream Dbn Model for Continuous Speech Recognition And Phone Segmentation *

Guoyun Lv, Dongmei Jiang, Pengjuan Guo, Ali Sun, Rongchun Zhao
Audio Visual Signal Processing Laboratory – AVSP: Northwestern Polytechnical University, School of Computer Science
Xi'an, Shaanxi, P.R. China
Email: lvguoyun101@sohu.com
H. Sahli, W. Verhelst
Vrije Universiteit Brussel, Dept. ETRO,
Brussels, Belgium
Email: hsahli @etro.vub.ac.be

ABSTRACT

HMM based models achieve promising performance in speech recognition. But for audiovisual bimodal speech recognition and phone segmentation, HMM based models, such as MSHMM and PHMM, lose in describing the asynchrony between audio and visual streams. Dynamic Bayesian Network (DBN) is expected to model the asynchrony well due to its flexible structure. As the first step, in this paper, we describe a single stream DBN model for continuous speech recognition and phone time segmentation. Word recognition rates and phone segmentation accuracies for 600 testing sentences are compared with those from HMM based speech recognition system (implemented by HMM toolkit HTK), Experiment results show that in clean and high SNRs environments, both DBN and HMM based models achieve similar performance, but in low SNRs environments, DBN has a better performance than HMM. At the same time, Phone segmentation is achieved by GMTK. These provide a foundation for using DBN based models in audiovisual bimodal speech recognition and asynchronous phone segmentation.

Keywords: Speech Recognition, HMM, Dynamic Bayesian Network, GMTK, Phone Segmentation.

1. INTRODUCTION

Speech and facial expression (such as lips, jaw, tongue, facial muscles, eyes, etc) are the important ways in the human intercourse. Both audio speech and visual speech are produced by the movements of articulator organs. There is an inherent asynchronous relationship between them. For a text to audiovisual speech synthesizer (TTAVS) system, how to determine the asynchrony between audio and visual speech, and reconstruct the vivid cartoon from well segmented audio and visual segments, is the active research topic of multi-model human-computer interface in many years. But some speech unit (such as phoneme, syllable, etc) is good for continuous speech recognition but not good for synthesizer, because it's difficulty to determine the asynchronous relationship between audio and visual speech.

In recent years, audio-visual speech integration, which takes full advantage of speech character information from both audio and visual streams, becomes an active topic for speech recognition and raise speech recognition rate under noisy acoustic environments [1,2,3]. For audio-visual speech recognition models, how to fuse the audio and visual features best is the main problem. Model fusion is the better way in previous research, the mostly used models are product HMM and multi-stream HMM (MSHMM) [4,5]. In these models, audio and visual streams are imported to two parallel HMM models with different topology structures respectively. But on some nodes, such as phone, syllable et al; constraints are imposed to limit the asynchrony between audio and visual streams to phone (syllable) level. Multi-stream HMM describes the correlation of audio and visual stream to some extent, and allows some asynchrony within phones. Comparing with single stream HMM, the system performance is improved especially in noisy speech environment. But it also has some problems [6]: 1) because of the large state space and the large amount of estimate parameters, the model become intractable. 2) It might cause the exponential weights on the stream probabilities to be improper (e.g., not summing to unity), and standard EM training algorithms cannot be used to estimate the stream exponential. 3) For large vocabulary continuous speech recognition task, multi-HMM can merely adopts phone to build models, so the model must be synchronized in phoneme state, but the asynchronous relationship of audio-visual exceeds phonemic boundary.

Because of the inherent limitation of MSHMM on asynchrony degree between audiovisual speeches, it cannot satisfy the need of asynchronous segmentation for visual speech synthesis. So the important thing is to find a new structure to describe the asynchronous relationship between audio and visual stream.

In recent years, the single stream and multi-stream DBN are applied to continuous speech recognition, and some research advances are achieved [7,8,9]. Based on the graphical model, DBN model has flexible structure. By building up the flexible model structure, Multi-stream DBN model breaks through the limitation of the multi-stream HMM. It may determine the asynchrony degree on word level for audio-visual streams. Different structures may be used between words, so asynchronous relationship of audio-visual stream can be faultlessly described.

But at present, most of DBN models are applied to continuous speech recognition [10,11,12], but don't pay attention to phone segmentation. That is to say, they just care for the word recognition rates, but ignore some interior complicated relation between audio and visual speech. This paper adopts a single stream DBN model to do continuous speech recognition and phone segmentation for speech

* Supported by the bilateral project between China MOST and the Flemish, Belgium: NO. [2004] 487. And the fund of 'The Developing Program for Outstanding Persons' in NPU—DPOP: NO. 04XD0102

under noise with various SNRs. The DBN based speech recognition is implemented by a graphical model toolkit (GMTK). Experimental data adopt 600 continuous digital speech, Recognition results are compared with HTK based speech recognition; For phone segmentation, both results from GMTK and HTK are compared with hand segmented results.

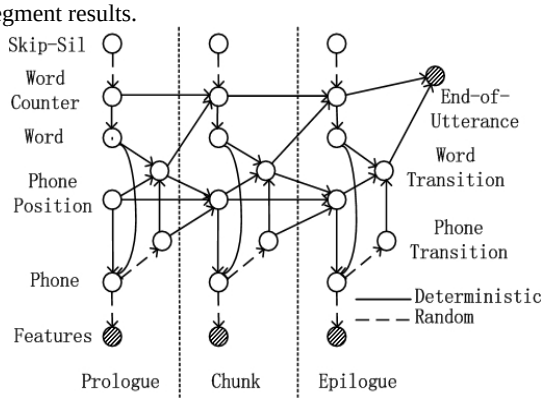
In section 2, the DBN model is described, section 3 describes experimental setup and results, and in section 4, some conclusions are drawn and the future work is discussed.

2. DBN MODEL

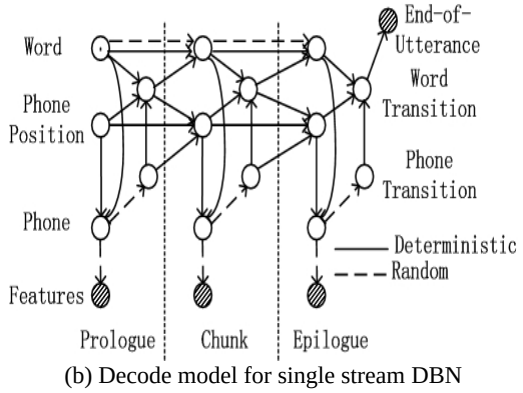
A Bayesian Network (BN) is a statistical model that can be use to describe collections of random variables and their dependencies. The DBN is used to model random variables as they evolve over time. DBN is a generalization of HMM, and HMM is a special case of general DBN. In DBN, all nodes expression is explicit; include word, phone, sub-word, etc. We use the DBN model structure for continuous speech given in [8] as our basic model, but change some interior meaning and conditional probability distributions (CPD) between nodes. We add phone table and mapping, and describe relationship of every word and its phoneme, change many corresponding script files and expression. The goal of our work is to make continuous speech recognition and phone segmentation by using the DBN model

2.1 Model description

Our DBN model is given in Fig.1, (a) is training model, (b) is the decode mode. They are the whole word models, the same as [8]. Based on the model of the document [8], we describe the relationship of every word and its phone in detail, increase a dictionary of word and phone, describe every nodes and responding conditional probability distribution (CPD). Finally make this model to print phone segment results.



(a) Training model for single stream DBN



(b) Decode model for single stream DBN

Fig.1 single DBN model for recognition and phone segment

In Fig.1, the DBN model consists of a prologue, a chunk to be repeated, and an epilogue. The prologue is the frame 0, and the epilogue is the last frame, a chunk can be expanding with the time slice. The node End-of-Utterance is the end of the sentence, which must be described in DBN model. The solid line denotes the deterministic relationship between the nodes, which must be described in the scrip file before training. The dash line denotes the random relationship between the nodes. In general, it is the dependency probability to be trained, which can be getting after the training; EM arithmetic is used in GMTK model training. As a training result, such as means, covariance and so on can be getting. The meaning of each node variable is as follows:

SKIP-SIL: A binary random variable, denoted as SS_t. Where *t* means the number *t* time slice.

Word Counter: the word position in current sentence, denoted as WC_t.

Word: the current word, determined by Word Counter and sentence, denoted as W_t

Word Position: the phone position index of the current word. It is denoted as WP_t.

Phone: be determined by a Word and the Phone Position, denoted as P_t.

Word Transition: describe when from a word to next word, denoted as WT_t.

Phone Transition: describe from a phone to next phone, denoted as PT_t

Feature: the observations stream feature, it adopts 39 dimension feature vectors, denoted as O_t.

End-of-Utterance: the end of the DBN, when it is 1, describe the end of the utterance.

Let the model T time slices, we may get the join probability distribution of the decode model, describe it with the follow formula.

$$\begin{aligned} &p(W_{1:T}, WT_{1:T}, PP_{1:T}, PT_{1:T}, P_{1:T}, O_{1:T}) = \\ &= \prod_{t=1}^{T} p(O_t|P_t)\, p(PT_t|P_t)\, p(P_t|WT_t, W_t)\, p(WT_t|W_t, PP_t, PT_t) \\ &\quad \times p(PP_t|WT_{t-1}, PP_{t-1}, PT_{t-1})\, p(W_t|W_{t-1}, WT_{t-1}) \end{aligned} \tag{1}$$

It is a complex structure, in order to infer and learn this model, we must determine some CPD, for fig.1, the nodes with shade describe the observation, and other nodes describe hide nodes. And the next step, we describe the CPD in detail.

2.2 Model nodes conditional probability distribution

For a better understanding, we describe the main nodes and their CPD (conditional probability distribution)

1) Feature vector O_t: P_t is the only parent of the O_t, describe it with $p(Q_t|P_t)$, it is similar as the HMM, use the mixture gauss model to describe it.

$$p(Q_t|P_t) = f(O_t|P_t) = b_{P_t}(O_t) \tag{2}$$

2) Phone Transition PT : it is a 0-1 distribution, it describe the end of one phone, and transfer to the next phone. It is same as the state conditional probability A_{ij}

3) Phone random *P*: it is described by WP_t and W_t with the follow formula.

$$p(P_t = j | W_t = i, WP_t = m) = 1$$
$$j = f(i, m) \quad (3)$$

I describe the word, m describe the phone position of the word.

4) Word Position WP_t: it has three parents, conditional probability is $p(WP_t | WP_{t-1}, WT_{t-1}, PT_{t-1})$, when word transition $WT_{t-1} = 1$, describe the end of the word, Word position is 0; when word transition $WT_{t-1} = 0$, it holds in the word, when phone transition is 1, word position add 1, or it is not change, and describe it with formula.

$$p(PP_t = j \mid PP_{t-1} = j-1, WT_{t-1} = m, PT_{t-1} = n) = \begin{cases} 1 & m = 1, j = 0 \\ 1 & m = 0, n = 1 \\ 0 & otherwise \end{cases} \quad (4)$$

5) Word Transition node WT, has three parents. Its conditional probability is $P(WT_t | W_t, WP_t, PT_t)$, for every word, we deal with it respectively. At first, when phone transition is 1, and WP_t is the last state of the word, word transition is 1. Otherwise, word transition is 0.

$$p(WT_t = j | W_t = a, PP_t = b, PT_t = m) = \begin{cases} 1 & j = 1, n = 1, a \in word(i), b = word(i, M) \\ 0 & otherwise \end{cases} \quad (5)$$

$word(i)$ M is the largest phone number of the word I.

6) Word node *W:* use the bi-gram model, when word transition is 0, the word holds; when the word transition is 1, the next word is generated according to the bi-gram.

7) Word Counter node *WC*: it is used in training model. In the frame 0, *WC* is 0. It has three parent nodes, the conditional probability is $p(WC_t \mid WC_{t-1}, SS_t, WT_{t-1})$.

If t-1 time slice word transition WT_{t-1} is 0, word counter holds. If t-1 time slice word transition WT_{t-1} is 1, check the word counter: If word counter is the word number of the utterance, then word counter does not changes; If not, check the silence SS: if SS_t is 0, word counter will plus 1; if SS_t is 1, word counter will plus 2. We describe it with the follow formula.

$$p(WC_t = j \mid WC_{t-1} = i, SS_t = m, WT_{t-1} = n) = \begin{cases} 1 & n = 0, j = i \\ 1 & n = 1, i = len(u), j = i \\ 1 & n = 1, i < len(u), m = 0, j = i + 1 \\ 1 & n = 1, i < len(u), m \neq 0, j = i + 2 \end{cases} \quad (6)$$

$len(u)$ Describe the length of the utterance.

3. EXPERIMENTS AND RESULTS

3.1 Database description and Feature Extraction

We use 200 continuous digits speech sentences recorded by a student to do the experiments. The sentences are the same as those in Aurora 2.0. The training set has 100 sentences; the other is the testing set. Noisy speech data is also made as testing sentences by adding gauss white noise with SRNs ranging is from 0db to 30db. The total testing set has 500 sentences.

MFCC features are extracted by HTK feature extraction program, with hamming window of 25ms, and frame shift of 100 frames per second. For each frame, 12 MFCC features and energy are extracted. Together with its delta and acceleration coefficients, totally 39 dimensional MFCC_D_A feature vectors are used.

The DBN based speech recognition is implemented by a graphical model toolkit GMTK. In which, we adopt the model given in fig.1, the same as [8], but change some interior meaning and conditional probability distributions (CPD) between nodes. We add phone table and mapping, and describe relationship of every word and its phoneme, change corresponding script files and expression, such as the number of the phoneme, feature vector dimension, and so on. At the same time, re-triangulate the training and decode structure file, and last, make the model to print phone segment and phone time alignment except the word sequence.

3.2 Setup

The first step is to test speech recognition rates and noisy-robustness of DBN, all clean speech data is divided into training set and testing set. Word recognition rates are compared for recognition results from GMTK and HTK under various SNRs. Experiments results are shown in Table 1. We can see that the recognition rates of DBN and HTK are similar for high SNRs testing data, but DBN model have the better performance than HMM for low SNRs data.

Table 1. Word recognition rates in various SNRs: (%)

Setup	0db	5db	10db	30db	Clean
HTK	18.36	37.20	57.97	93.91	96.96
DBN	35.85	69.81	87.74	98.11	99.06

The second step is to make phone segmentation and time alignment. Based on the step 1, both the segmented phone time boundaries from GMTK and HTK are compared with hand segmented phone time boundaries. So far, there is not the better evaluation criterion for phone segmentation. For evaluating the accuracy of segmentation, we propose the following objective standard.

Suppose that a sentence consists of *N* phones, for handiwork phone segmentation, the boundary time of phone is from a_i to b_i, a_i is the start time and b_i is the end time. In the similar way, suppose phone segmentation results from GMTK is from c_i to d_i, and phone segmentation results from HTK is from e_i to f_i, so we can achieve the follow evaluation formula:

For GMTK, we use the following objective standard to evaluate the phone segmentation error, denoted as *GMTKSE*:

$$GMTKSE = \frac{\sum_{i=1}^{N} (|c_i - a_i| + |d_i - b_i|)}{N} \quad (7)$$

In which N is phone number of the sentence.

For HMM, the objective standard *HMMSE* is as follows:

$$HMMSE = \frac{\sum_{i=1}^{N}(|e_i - a_i| + |f_i - b_i|)}{N} \quad (8)$$

In which N is phone number of a sentence.

For example, for the sentence “sil five four sil”, we list the phone segment results in table 2.

Table 2: phone segmentation for sentence “sil five four sil”

	Sentence: sil five four sil		
	HTK (ms)	Handiwork (ms)	GMTK (ms)
Sil	0-290	0-290	0-290
F	290-360	293-370	300-400
ay1	360-510	370-500	410-530
V	510-750	500-710	540-680
F	750-820	710-820	680-790
ao1	820-1070	820-1060	780-1060
r	1070-1150	1060-1120	1070-1110
sil	1150-1620	1120-1620	1120-1610

We may get the phone segmentation error of the whole sentence according to formula Eq. (7) and Eq. (8): the results are 25ms for HTK and 43ms for GMTK.

Further, we adopt the follow phone segmentation evaluation formula for the whole group of sentences, denoted as *GMTKWSE* for GMTK *and HMMWSE* for HMM.

$$GMTKWSE = \frac{1}{M}\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{N_j}(|c_i - a_i| + |d_i - b_i|)}{N_j}\right) \quad (9)$$

$$HMMWSE = \frac{1}{M}\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{N_j}(|e_i - a_i| + |f_i - b_i|)}{N_j}\right) \quad (10)$$

In which *M* is the number of whole group of sentences, and N_j is the phone number of the j-th sentence.

Acceding to formula Eq.(9) and Eq.(10), we get phone segmentation error of the 100 clean speech testing data, result for GMTK is 65ms, and that for HTK is 48ms.

From the results in table 1 and the whole phone segmentation error, we can see that GMTK and HTK have similar performance on phone segmentation for various SNRs speech data.

4. CONCLUSION

In this paper, we describe our initial work on a single stream Dynamic Bayesian Network model for speech recognition and phoneme segmentation. Results show that this DBN model has the better performance than HMM for continuous speech recognition, at the same time, the DBN has the similar performance as HMM for phoneme segmentation. That is to say, we can use the GMTK to make continuous speech recognition and phone segmentation, and get the phone sequence include time boundary. So this important conclusion provides the foundation for building audio-visual bimodal multi-stream DBN model, and to determine the phone asynchronous relationship of audio-visual features in continuous speech. In the future work, by adopting words as the synchronous nodes and loosing the degree of asynchrony limitation between audio and visual features, we plan to construct the multi-stream DBN model for speech recognition and bimodal phone segmentation, and try to find the better asynchrony relationship between audio and visual speech. It’s very important for building up text to audiovisual speech synthesizer (TTAVS) system.

5. REFERENCES

[1] G. Potamianos, C. Neti, et al, Recent advances in the automatic recognition of audiovisual speech. Proc. IEEE, 2003, 91 (9): 1306 – 1326.

[2] A.V. Ne_an, L. Liang, et al, A coupled HMM for audio-visual speech recognition, in Proc. of IEEE International Conf.on Acoustics, Speech and Signal Processing, Orlando, Florida.

[3] C. Neti, G. Potamianos, et al, and Large-vocabulary audiovisual speech recognition: A summary of the Johns Hopkins Summer 2000 Workshop, in Proc. Works. Signal Processing, 2001.

[4] Bibliographic reference. Nigerian, Mukundh / Sreenivas, T. V. (2003): "Product-HMM - a novel class of HMMs for sub-sequence modeling", In WSLP-2003, 117-124.

[5] A.Hagen, A. C. Morris ， Recent advances in the multi-stream HMM/ANN hybrid approach to noise robust ASR IDIAP-RR 02-57, 2002.

[6] M. Molander. “Experiment with Asynchrony in Multimodal Speech Communication”, Mater Degree Thesis in KTH Speech, Music and Hearing Lab, 2004.

[7] Jeff Bilmes， GMTK: The Graphical Models Toolkit， http://ssli.ee.washington.edu/~bilmes/gmtk/

[8] J. Bilmes, G. Zweig, et al, “Discriminatively Structured graphical models for speech recognition, JHU-WS-2001 final workshop report,” Johns Hopkins Univ., Baltimore, MD, Tech. Rep. CLSP, 2001, [Online]. Available, http://www.clsp.jhu.edu/ws2001/groups/gmsr/GMRO-final- rpt.pdf.

[9] J. Bilmes and G. Zweig, ìThe Graphical Models Toolkit: An open source software system for speech and time-series processing, in Proc. ICASSP, 2002.

[10] Y.M. Zhang, Q. Diao, et al, “DBN based multi-stream models for speech”, IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 2003.

[11] S. Amarnag, J.N. Gowdy, et al, "DBN based Multi-Stream Models for Audio-Visual Speech Recognition", To Appear in Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing, Montreal, 2004.

[12] K. Saenko, K. Livescu, et al, Production domain modeling of pronunciation for visual speech recognition, in Proc. ICASSP, 2005.

Fractal Image Compression using a Two-Pass Improved Encoding Scheme

Ching Kin Wah Eugene
School of Computing, National University of Singapore
Republic of Singapore
Email: chingkin@comp.nus.edu.sg
Ong Ghim Hwee
School of Computing, National University of Singapore
Republic of Singapore
Email: onggh@comp.nus.edu.sg

ABSTRACT

An improvement scheme, so named the Two-Pass Improved Encoding Scheme (TIES), for the application to image compression through the extension of the existing concept of fractal image compression (FIC), which capitalizes on the self-similarity within a given image to be compressed, is proposed in this paper. In this paper, we first briefly explore the existing image compression technology based on FIC, before proceeding to establish the concept behind the TIES algorithm. We then devise an effective encoding and decoding algorithm for the implementation of TIES through the consideration of the domain pool of an image, domain block transformation, scaling and intensity variation, range block approximation using linear combinations, and finally the use of an arithmetic compression algorithm to store the final data as close to source entropy as possible. We then conclude by explicitly comparing the performance of this implementation of the TIES algorithm against that of FIC under the same conditions.

Keywords: Fractal Image Compression, Quad-tree Partitioning, Peak Signal-to-Noise Ratio, Arithmetic Coding, Linear Combination.

1. INTRODUCTION

Fractal Image Compression

Fractal image compression is a lossy compression technique proposed in the 1980s by Michael Barnsley [1-3]. The technique is based on the observation that since fractals can generate relatively realistic images, then, the converse, which is the ability to store a given image in the form of just a few basic fractal patterns, coupled with the specification on how to use those fractal patterns to restore the image should also be possible. The FIC algorithm starts from the complete image, and breaks down that image into a number of partitions. For each given partition P_i, the algorithm then searches the image for other sub-sections of the image, S_j, which are relatively similar to P_i, and then maps P_i to S_j. Once the algorithm has found a mapping for every sub-section of the image, it terminates. The compressed image would then consist of all the partitions P_i, and their corresponding mappings, called the "code book" of the image. In FIC terminology, P_i is called a domain block, and the corresponding set of partitions is called the domain pool, while S_j is called a range block, and the corresponding set of sub-sections is called the range pool.

An Improvement Scheme to FIC

This paper proposes an improvement scheme which is based in part on the earlier described FIC algorithm, which will reduce the overall size of a compressed image as compared to that of FIC under similar conditions [3]. This scheme will, in fact, make extensive of use the FIC algorithm for partitioning a given image into range blocks (sub-images of the original image) and extracting the domain pool by finding a suitable mapping to approximate the range blocks from the domain blocks. However, we note that FIC searches for the best match between a range block and a domain block, and then proceeds to store that mapping range block to domain block. In other words, FIC produces a one-to-one mapping for every range block from a set of domain blocks. This has two consequences. Firstly, the domain to range mapping is the best for a one-to-one mapping, but would it be possible that a linear combination of domain blocks might result in a better approximation of that range block? If so, then image quality can be increased without increasing the size of the domain pool. In addition, by using linear combinations of domain blocks to approximate the range blocks, it also becomes possible to reduce the size of the domain pool should some of the domain blocks now be made redundant. This is similar to the idea of using a resultant vector to represent a linear combination of vectors in the technique of matching pursuit [4]. Secondly, can the size of the domain pool be reduced if we search for similar domain blocks within the domain pool itself?

Indeed, both of these are possible, and this paper proposes an extension and improvement to the original FIC algorithm by capitalizing on these two observations to improve compression performance. However, it is worthwhile to note that while the idea behind searching for linear combinations of domain blocks instead of simply using one-to-one mappings of domain blocks to range blocks seems simple, determining what is considered the best linear combination is extremely computationally expensive, as the algorithm would have to search through all possible linear combinations of varying sizes. Thus, this paper will also propose a simple and effective algorithm to searching for such linear combinations.

2. THE PROPOSED ALGORITHM

Introduction

The proposed algorithm is a two-pass scheme, whereby the first-pass involves extracting the domain pool (as in the original FIC algorithm), and the second-pass involves utilizing that extracted domain pool to achieve the maximum compression. Due to its two-pass nature, the proposed algorithm is thus named as a "Two-Pass Improvement Encoding Scheme" (TIES). The first-pass of the TIES algorithm is straight-forward as it derives directly from the FIC algorithm. In the second-pass, the TIES algorithm performs:

Compression of the Domain Pool
Partitioning of the image
Searching for the best linear combinations
Storage of the results

In the presentation of the TIES algorithm in this paper, a square grayscale image is assumed in order to provide a simple framework for the description and implementation of the algorithm. The algorithm can be extended easily to accommodate non-square image by first dividing the non-square image into two parts, the largest possible square I_{sq1}, and the remaining part of the image I_{r1}, and then applying the algorithm on I_{sq1}. By recursively applying the above steps of dividing I_r again into I_{sq2} and I_{r2}, it is clear that we will finally be able to apply the compression algorithm on the entire image. In addition, the algorithm can also be easily extended to colour images. Colour images are 24-bit images consisting of the layers red, green and blue (for RGB images). By first breaking the colour image into its 8-bit components (i.e., red, green and blue layers), and then applying the compression algorithm to each layer separately, and finally reconstructing the complete image at the decoder level, we can use the algorithm, without modification, on colour images. Hence, since the algorithm described in this paper can be relatively easily extended to non-square, colored images, it suffices to describe the algorithm for a square, grayscale image for clarity.

Quad-Tree Partitioning

A simple way to partition an image is simply to break the given image up into fixed-sized range blocks, R_i. However, such a method of partitioning has a weakness – there are some parts of an image where there is less detail (for example, a background scene). Hence, larger range blocks will suffice to cover that area well. This will, in turn, reduce the number of domain blocks needed to cover the image as a whole, as well as the number of domain to range block mappings, thus achieving better compression. Likewise, there are also other regions of that same image, which are difficult to cover well using a range block of fixed size. Such regions usually require smaller range blocks in order to capture the finer detail of that portion of the image (for example, the eyes of a person).

Hence, to allow for varying range block sizes, quad-tree partitioning is used in the implementation of the TIES algorithm. In quad-tree partitioning, a square in the image is sub-divided into four equally sized squares when it is not sufficiently well covered. The measurement of well coveredness is determined by the tolerance factor of the TIES encoder, and will be fully detailed in Section 3. This process repeats recursively starting from the original image (assumed to be a square as explained above), and continuing until a given square (the range block) is small enough to be well-covered by domain block(s). Small squares can be covered better than large ones because contiguous pixels in an image tend to be highly correlated [2, 3].

3. THE TIES ENCODER

Extracting the Domain Pool

The extraction of the domain pool is similar to the inference algorithm used by the FIC encoder. The original image is first divided into a number of overlapping sub-squares called D_i of size 2^n by 2^n, where n = 0, 1, … , $\log_2$(size of image). The collection ***D'*** of these sub-squares forms the initial of the set of domain blocks, or the initial domain pool. Next, the original image then partitioned, using quad-tree partitioning, into four equally sized sub-sections R_i called range blocks. For each R_i, the algorithm tries to find a best match of D_i from ***D'***, such that the tolerance criteria is met. If such a match is found, then a mapping is made between D_i and R_i, and D_i is placed into another collection ***D***, which contains the index of the corresponding domain block in ***D*** chosen during the extraction process. ***D'*** thus is the set of indices representing the final domain pool. If a match cannot be found, then R_i is again sub-divided using quad-tree partitioning into R_{i1}, R_{i2}, R_{i3} and R_{i4}, and for each R_{ij}, where j = 0, 1, 2, 3, the algorithm is again applied. This recursive process finally terminates when all R_i have a mapping to a certain D_i, and the final collection ***D*** is the domain pool.

Compressing the Domain Pool

Once the domain pool ***D*** has been extracted, we then proceed to compress the domain pool by searching for all domain blocks, which are similar.

The key in this algorithm is to find all domain blocks which are similar. Two domain blocks, D_a and D_b, are considered to be similar if the sum of the square of the difference in pixel values between corresponding pixels in D_a and D_b is less than a specified tolerance T, for all possible 90^o rotations and flips applied to D_b. This means that the comparison between D_a and D_b has to be applied a total of eight times, four times for each 90^o rotation of D_b without flipping D_b, and another four times with a horizontal flip applied to D_b. Clearly, when making comparisons between D_a and D_b, we also have to ensure that D_a and D_b are of the same size.

Searching for the Best Linear Combination

One of the issues in finding the best linear combination is the fact that exhaustively searching through all possible combinations for all possible sizes is computationally expensive, and hence, doing such an exhaustive search would clearly make the encoding process excessively long. Hence, this paper will propose an algorithm that can be implemented simply and efficiently.

The proposed algorithm is implemented by restricting that the reconstruction of the final range block R from its corresponding linear combination of domain blocks $\boldsymbol{L} = \{D_1, D_2 \ldots D_n\}$ is given by taking the sum across all elements of L (i.e., $R = D_1 + D_2 + \ldots + D_n$). On each iteration, the algorithm then computes,

$$\text{Remainder} = \text{R} - \sum_{i=0}^{n} D_i \qquad (1)$$

and continues to iterate infinitely, each time choosing another domain block D_{i+1} from ***D*** to add to the previously computed sum of domain blocks, but only if the addition of the next domain block will bring the remainder closer to zero. If no such domain block D_{i+1} can be found, the algorithm terminates for this particular range block R. In addition, the algorithm will also terminate when the remainder is less than a tolerance value T_{LC}. Thus, this tolerance value represents whether a given range block is adequately represented by its linear combination of domain blocks (i.e., well-covered).

When performing the comparison between domain blocks D_{i+1} to choose the best domain block, the algorithm also applies two flip transformations, four rotation transformations, and a number of intensity multipliers on each domain block. Visually, the intensity multiplier simply lightens the image associated with the domain block, such that $D'_{i+1} = I \cdot D_{i+1}$, where I is the intensity multiplier, and D'_{i+1} is the modified domain block.

In addition, instead of taking only the sum of the chosen set of domain blocks in the linear combination to represent a given range block, an alternative, allowing the linear combination search algorithm to represent a given range block using both the sum and difference of individual domain blocks. It can be seen that such an arrangement would allow the linear combination search algorithm much more flexibility in obtaining a good representation of the range block. It is, in fact,

likely, that such an arrangement would allow the encoder to increase image quality, with only a small impact on compression performance.

Storage Format

Since the set of domain blocks ***D'*** has been chosen, the original domain pool ***D*** can be further reduced in size to contain only those blocks which are elements of ***D'***. In other words, we want ***D*** = ***D'***. Hence, we now perform a second level compression of the original domain pool ***D*** to restrict ***D*** to elements of ***D'***, and we also adjust the pointers in ***D'*** such that they point at the correct (changed) element in ***D***. This hence reduces the size of ***D***. At this point, the TIES encoder would have obtained:

- ***D:*** Compressed domain pool
- ***D':*** Set of chosen domain blocks (pointers to blocks in D)
- ***F':*** Set of corresponding flip transformations used
- ***R':*** Set of corresponding rotation transformations used
- ***I':*** Set of corresponding intensity multipliers used
- ***LC':*** Set of corresponding number of linear combination elements and the range block size

Thus, it is now necessary to obtain an efficient encoding of the data to minimize the final output file size. First, a minimal bit coding scheme will be used to pack the data efficiently by observing the limits on the type of data stored. Secondly, the data will be further compressed using arithmetic encoder [5]. Sets F' and R' can be combined into a single set T' comprising of 3-bit elements. This allows for more efficient compression during arithmetic coding.

We note that at this point there has been no effort made to store the coordinates (location) of each range block in the original image. This omission of the location of each range block is intentional for the purposes of reducing the size of the compressed data. However, the location of each range block can be determined, due to the way in which the data blocks LC', T' I' and D' are arranged. The sequence in which the data for LC', T', I' and D' is written into their corresponding data blocks is such that they represent, sequentially, the range blocks from top to bottom, left to right, regardless of the size of the range block itself. This allows the decoder to correctly retrieve the locations of each range block. The technique and corresponding algorithm used in handling the varying sizes of the range blocks will be described in Section 4.

Finally, after the 5 blocks of data (D, D', T', I', LC') have been constructed, a (block-based) arithmetic coder is used to further compress each individual data block to achieve maximum possible compression.

In addition, due to the highly independently nature of the encoding algorithm, in particular the linear combination search algorithm, it is also possible to construct a parallelized variation of the algorithm in order to achieve improved performance [6-8].

4. THE TIES DECODER

The implementation of the TIES decoder is fairly straightforward as compared to the encoder, with the only slightly more challenging task being the means to decode the range block location based on the encoding format. Essentially, the decoder reads the encoded file, extracts from the five encoded blocks the matrices and storing them back into the sets ***D***, ***D'***, ***T'***, ***I'*** and ***LC'*** by implementing an arithmetic decoder, inferring the range block locations and storing them in sets $\boldsymbol{R}_x$ and $\boldsymbol{R}_y$, and finally reassembling the final image using the reconstructed information.

With the seven matrices ***D***, ***D'***, ***T'***, ***I'***, ***LC'***, $\boldsymbol{R}_x$ and $\boldsymbol{R}_y$, the decoder now has sufficient information to reconstruct the image. For each range block, the decoder obtains the (x, y) location of that range block from $\boldsymbol{R}_x$ and $\boldsymbol{R}_y$, and regenerates the range block by applying the correct transformations to each used domain block in ***D***. Finally, with the reconstructed image, the decoder writes the data out in BMP format.

Once the decoder has obtained the final image in terms of raw pixel values, the decoder then performs post-processing and smoothing to remove any blocking artifacts found due to the partitioning of the image into discrete range blocks during the encoding phase, as well as to visually enhance the image. Upon completion of the post-processing, the image is written to an output file in BMP format.

5. THE PSNR METRIC

In measuring the compression performance of the TIES algorithm, as well as that of the FIC algorithm, it is useful of have a precise and formal method of measuring image quality, rather than relying purely of visual inspection of the final decoded image. Hence, this paper makes use of the peak signal-to-noise ratio (PSNR) as a representative of image quality.

$$\text{PSNR} = -20 \log \frac{\sum_{p=0}^{total\ pixels} \left| image_{a,p} - image_{b,p} \right|}{total\ pixels \times 255} \text{ dB} \quad (2)$$

6. RESULTS

In evaluating the performance of the TIES encoder, an empirical analysis was carried out on six grayscale test images. These test images were chosen as representatives of certain broad classifications of image types, which include human subjects, man-made objects, natural scenery and combinations of the above mentioned types.

During the empirical analysis, two separate encodings, one by the TIES encoder, and the other by the FIC encoder [3], were performed on a total of four different image sizes – 128x128px, 256x256px, 512x512px and 1024x1024px – for each of the six test images. The number of linear combination elements that the encoder was allowed to use was limited to four. The reason for this is that statistically, four elements tend to produce a suitable balance between image quality and image compression. The statistics that were captured include:

1. Savings due to domain compression in the TIES encoder
2. Savings due to arithmetic coding in the TIES encoder
3. Compression performance of the TIES vs. FIC encoders
4. Impact of the linear combination search algorithm
5. Encoding and Decoding times of the TIES vs. FIC encoders

The six test images used in this empirical study are:

Finally, in the course of the empirical analysis carried out, we note that in implementing the linear combination search algorithm, only the sum of *n* domain blocks are allowed to

represent a range block.

Compression Performance of TIES vs. FIC

In this section, the compression performance of the TIES encoder in comparison to the FIC encoder is examined. In Fig. 6.1. below, the graph of compression performance vs. PSNR is shown for the Lena image. The compression performances for the remaining images are given in Tables 6.1 and 6.2. In this analysis, compression is defined as the percentage savings obtained in compressing a given image. Mathematically:

$$\text{Percentage Savings} = \frac{\text{Size of original image - Size of compressed image}}{\text{Size of original image}} \times 100\% \quad (3)$$

Graph of Compression Performance (y-axis) vs. Image Size (x-axis) for Lena Image

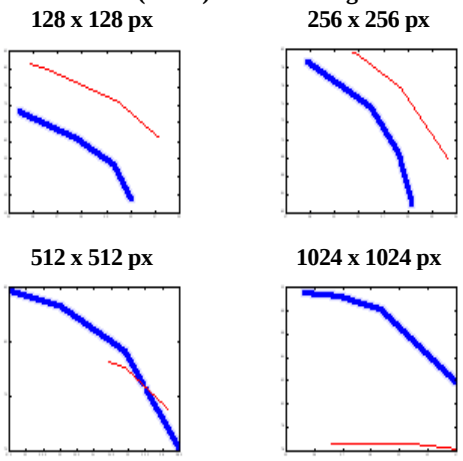


Fig. 6.1. Compression performance of TIES (thick) vs. FIC (thin)

Table 6.1(a). Size of large (1024x1024px) compressed file using TIES and FIC encoders for remaining images

Images of Size 128 x 128 px *(+: high PSNR, -: low PSNR)*								
	Boat		**Bike**		**Plant**		**Lady**	
PSNR	–	+	–	+	–	+	–	+
FIC	**54**	**81.4**	**81.3**	**57.6**	**81.3**	**53.6**	**82.3**	**53.7**
TIES	44.8	62	63.2	40.1	61	34.8	61.2	32.9

Table 6.1(b). Size of small (128x128px) compressed file using TIES and FIC encoders for remaining images

Images of Size 1024 x 1024 px *(+: high PSNR, -: low PSNR)*								
	Boat		**Bike**		**Plant**		**Lady**	
PSNR	–	+	–	+	–	+	–	+
FIC	76.6	76.5	76.9	74.4	76.6	75.1	74.6	72.2
TIES	**89.7**	**82.1**	**88.1**	**76.1**	**87.5**	**77.9**	**87.7**	**78.9**

From Fig. 6.1. and Table 6.1., we observe that with small images, the compression performance of the FIC encoder is better than the TIES encoder regardless of image quality (PSNR). However, when the size of the encoded image increases to 1024x1024px, the TIES encoder consistently produces better compression ratios compared to the FIC encoder at equal or better PSNR values. Table 6.2. below summarizes the specific improvement in compression performance of TIES over FIC for 1024x1024px images.

Table 6.2. Compression performance gains for TIES over FIC for large (1024 x 1024px) images

Image	**PSNR Range**	**% Improvement in Compression over FIC**		
		Min	**Max**	**Mean**
Lena	37 – 41	6.0	12.5	9.25
Boat	37 – 40	5.5	13.0	6.75
Bike	35 – 37.5	2.5	6.0	5.5
Plant	33 – 37.5	2.5	11.0	8.375
Lady	33 – 37.5	7.0	13.0	10

From the table above, we observe that within a given overlapping range of PSNR values produced by the TIES encoder and the FIC encoder, the TIES encoder achieves greater savings in the range of between 3 and 8.5 percentage points over the FIC encoder. As such, this paper claims that while the TIES encoder produces smaller savings on average when compared to FIC for small-sized images, TIES will outperform FIC as image size is increased.

In the previous analysis, we have excluded a discussion on the results of the sky image. The reason for this is that it can be clearly seen that the encoding for this image appears strange. In fact, the resultant encoding based on the TIES encoder produces extremely high compression ratios (88% to 98%) but poor PSNR values (17 to 19) across *all* image sizes. Correspondingly, the FIC encoder produces high PSNR values (39 to 52) but average compression (80% to 82%). As a result, the TIES decoded image suffers from severe impairment after being decoded. Thus, in this case, the TIES encoder failed to perform satisfactorily. A possible reason for this is discussed in Section 6.

Savings due to Domain Compression (TIES)

In Fig. 6.2. below, the savings produced by the TIES encoder due to domain compression (only) is shown for each of the seven individual test images. The horizontal axis of the graph represents the size of the test image, while the vertical axis represents the percentage savings obtained in compressing the original domain pool (extracted by the FIC algorithm). Mathematically:

$$\text{Percentage Savings} = \frac{\text{Original domain pool size - New domain pool size}}{\text{Original domain pool size}} \times 100\% \quad (4)$$

Graph of Percentage Savings (y-axis) vs. Image Size (x-axis)

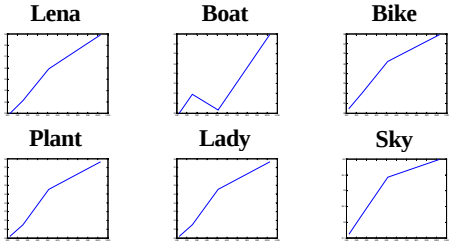


Fig. 6.2. Percentage savings due to domain compression

From the graphs above, two observations can be made. Firstly, for all six images, a reduction in domain pool size is

achieved by compressing the domain pool. Furthermore, from the statistics captured, it appears that the size of the compressed domain pool is in the range of 1% to 11% of the size of the original domain pool.

Secondly, we observe that for the test images, generally the percentage savings due to domain pool compression increases as the size of the test image increases. As such, this paper suggests that there is a positive, linear relationship between the size of the original image and the percentage savings gained due to domain pool compression. Indeed, an abnormality to this trend can be observed in the boat image, where it appears that the 512x512px image resulted in fewer saving as compared to the smaller 256x256px image. However, since the domain pool extracted is dependent on the nature of the image, and since the difference in percentage savings between the 256x256px image and the 512x512px image is less than 2%, it is likely that the abnormality is primarily due to statistical fluctuations of the compressed domain pools, and that the general trend of the savings due to domain compression is still linear with respect to an increase in the size of the image encoded.

Finally, we observe that for the final image, the percentage savings achieved is extremely high. However, as described in the previous section, the PSNR value for this image is extremely low. Table 6.3. shows size of the original domain pool and the size of the compressed domain pool for this image.

Table 6.3. Difference between size of original and compressed domain pools

Image Size (px^2)	Domain Pool Size					
	Average of Images 1 to 5				Sky Image	
	Orig.		Compressed		Orig.	Com-pressed
	Size	Deviation (%)	Size	Deviation (%)		
128	2322	7.81	202	13.09	1098	26
256	6742	8.81	488	8.75	4152	32
512	20451	6.23	1034	12.41	19389	30
1024	70260	3.93	1204	16.46	65559	11

As can be seen from the table, the size of the compressed domain pool of the final image is in the order of 10^1, while the average size of the compressed domain pool for the first six images is in the order of 10^2, one full difference in magnitude. The key difference between this image and the first six images is the homogenous nature of this image. This would mean that in the extracted domain pool, a large number of blocks would be relatively similar to each other, which causes the TIES encoder to decide that a large proportion of domain blocks can be made redundant and thus removed from the domain pool. However, due to this excessive reduction of the domain pool, the TIES encoder in its search for linear combinations to represent each range block is unable to obtain such a linear combination from the extremely limited domain pool, resulting in a poor approximation to the original image. This thus explains the poor PSNR of the decoded image.

Savings due to Arithmetic Coding (TIES)

In Fig. 6.3. below, the savings produced by the TIES encoder due to arithmetic coding (only) is now shown for each of the six test images. The horizontal axis of the graph represents the size of the test image, while the vertical axis represents the percentage savings obtained in encoding each of the five data segments (***D***, ***D'***, ***LC'***, ***T'***, ***I'***) using the block-based arithmetic coder. Mathematically:

$$\text{Percentage Savings} = \frac{\text{(Original size - Compressed size) of data segments}}{\text{Original size of data segments}} \times 100\% \quad (5)$$

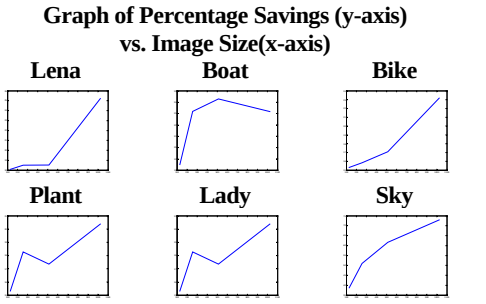
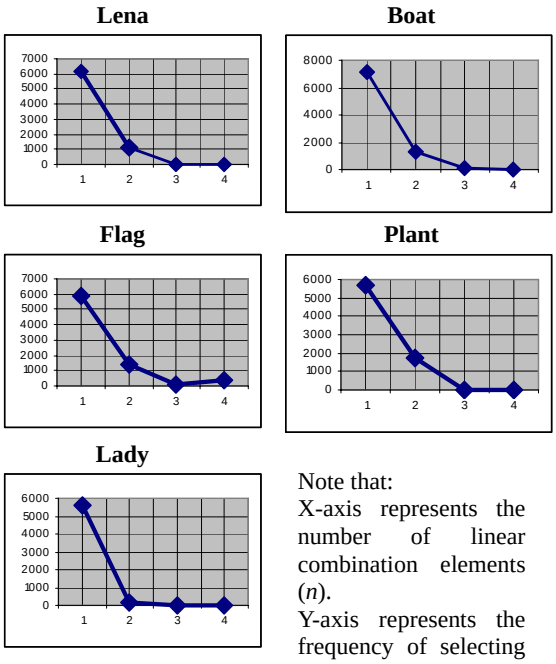


Fig. 6.3. Percentage Savings due to Arithmetic Coding

Again, we observe that the empirical results suggest that savings between the range of 15% to 26% can be achieved by encoding the data blocks using an arithmetic coder [5]. In addition, this paper again suggests that the relationship between the size of the image being encoded, and the percentage savings achievable due to arithmetic coding is linear and positively correlated. Indeed, this is in line with the results of the well-understood arithmetic-encoding algorithm, where data sets whose elements have asymmetric frequencies can be compressed significantly.

Impact of Linear Combination Search Algorithm

In this section, we perform a more detailed analysis of the linear combination search algorithm, observing the frequency at which the algorithm selects *n* domain blocks to represent a given range block.



Note that:
X-axis represents the number of linear combination elements (*n*).
Y-axis represents the frequency of selecting *n* domain blocks.

Fig. 6.4. No. of domain blocks used by linear combination search algorithm

From Fig. 6.4. above, it can be observed that linear combination search algorithm reveals that the algorithm, on average, selects 1 linear combination element to represent the range block 82.9% of the time, and selects 2 linear combination elements 15.5% of the time. The remaining range blocks are represented using either 3 or 4 linear combination elements. As such, there appears to be a negatively exponential relationship between the numbers of linear combination elements chosen for a given image.

Encoding and Decoding Times of TIES vs. FIC

In Fig. 6.5., the average encoding and decoding time for each image size, across all six test images were taken for both the TIES encoder and the FIC encoder is shown. The horizontal axis of the graph represents the size of the image being encoded, while the vertical axis represents the time taken in seconds for the encoding process to complete.

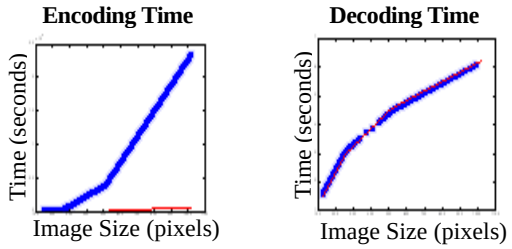


Fig. 6.5. Encoding and decoding times of TIES (thick) and FIC (thin)

It can be observed from Fig. 6.5. above that for small images (256x256px and below), TIES and FIC complete the encoding process in approximately the same amount of time. However, as the image size increases, the graph above suggests that the encoding time for TIES increases much faster than that of FIC. However, this is to be expected as there usually exists a trade-off between time and space. Thus in the case of the TIES encoder, increased compression is achieved at the expense of encoding time. The reason for the exponential growth in encoding time for the TIES encoder is due to the complexity of the domain compression linear combination search algorithms.

In the case of decoding, it can be observed that the decoding time for TIES is comparable to that of FIC. In this light, we note that the TIES algorithm, similar to the FIC algorithm, is an asymmetric encoding/decoding algorithm.

7. CONCLUSION

This paper has presented a proposed improvement scheme (Two-Pass Improved Encoding Scheme) based on the concept of fractal image compression. In describing the TIES encoder and decoder, three algorithms are used, which forms to basis of the TIES encoding algorithm – domain pool compression, best linear combination search, and a block-based arithmetic coder. To exemplify the workability of TIES, as well as to demonstrate that, under similar conditions, the TIES algorithm does outperform its FIC counterpart for large images of the size of 1024x1024 pixels and above, we have derived a complete implementation of the TIES encoder and decoder, and obtained empirical results based on seven test images.

From the results of the empirical analysis carried out in Section 6, we have observed that the TIES encoder produces good results in terms of gaining additional compression over the FIC encoder for large images of size 1024x1024px. In addition, there appears to be a linear relationship between the size of the image encoded and the savings due to (1) domain pool compression and (2) arithmetic coding. Based on these observations, this paper thus suggests that there is indeed a linear relationship between the size of the image encoded and the overall compression performance of the TIES encoder. In other words, this paper suggests that as the image size increases, TIES will out-perform the FIC encoder by an increasing margin. In addition, we note that for the sample images used in the empirical analysis and the results obtained, the TIES encoder is able to achieve greater savings in the compressed file size between 2.5 and 13 percentage points with reference to the size of the original image, and achieves better performance in the range of 3 to 8.5 percentage points with respect to the FIC encoder, for large images.

Finally, we established that the current implementation of the TIES encoder has a drawback – it is unable to handle largely homogenous images, reason being that the domain compression algorithm over-reduces the original domain pool extracted to the point that it is inadequate for the linear combination search algorithm to find a suitable linear combination of domain blocks to represent each range block in the image.

8. ACKNOWLEDGEMENTS

This authors' work was supported by the National University of Singapore Academic Research Fund under grant R-252-000-118-112.

9. REFERENCES

[1] A. E. Jacquin, “Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations”, IEEE Transactions on Image Processing, 1992, 18-30.

[2] F. Yuval, ‘Fractal Image Compression”, SIGGRAPH ’92 Course Notes, San Diego Super Computer Center, University of California, San Diego, USA, 1992.

[3] Y. Fisher (editor), “Fractal Image Compression – Theory and Applications”, Springer-Verlag, New York, USA, 1995.

[4] S.G. Mallet, Z. Zhang, “Matching Pursuits with Time-Frequency Dictionaries”, Journal of IEEE Signal Processing, 41（12）, December 1993.

[5] I. H. Witten, R. M. Neal, J. G. Cleary, “Arithmetic Coding for Data Compression”, Communications of the ACM, 30 （6）, 1987.

[6] J.Hammerle, A.Uhl, “Fractal image compression on MIMD architectures II: Classification based speed-up methods”, Journal of Computing and Information Technology (Special Issue on Parallel Numerics and Parallel Computing in Image Processing, Video Processing, and Multimedia), 2000, 8(1):71-82.

[7] Shinhaeng Lee, Shin'ichiro Omachi and Hirotomo Aso, “A Parallel Architecture for Quadtree-based Fractal Image Coding”, Proceedings of the 2000 International Conference on Parallel Processing (ICPP2000), August 2000，15-22.

[8] M. Wang, R. Liu, et al, “Adaptive Partition and Hybrid Method in Fractal Video Compression”, to appear in a special issue of Computers and Mathematics with Applications Distributed Algorithms for Science and Business, 2006.

Color Video Compression Using Fractal Method

Sihui Chen
College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian 350002
Email: centrychen@hotmail.com
Meiqing Wang
College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian 350002
Email: mqwang@fzu.edu.cn

ABSTRACT

Fractal video compression is a relatively recent image compression method. In this paper we proposed two color video compression algorithms using fractal method. In first method a color video with RGB mode is firstly split into three channel components, and then these components are combined to a long sequence, which is then compressed by fractal gray video compression method. In second method a color video with RGB mode is converted to one with YIQ mode. Components Q and I are shrunk to a quarter size of original ones respectively; and then these components are compressed with fractal gray video compression method respectively. Experimental results show that the first method can get a high reconstruction quality and the second one can achieve a high compression ratio without perceptible image degradation.

Keywords:
Fractal image compression, Gray-level video compression, Color video compression

1. INTRODUCTION

Data compression has become an important issue in information storage and transmission. This is especially true for the environment which consist of a large number of video images, such as multimedia application, digital library and computer-aided design, etc.. Recently, a large quantity of methods has appeared in literature for achieving high compression for video compression, such as MPEG [1], H.263 [2], JPEG [3] and so on. Among them, the fractal method is a feasible and promising compression technique.

Up to the present, fractal video image compression research is mainly performed on gray-level video images, with relatively few results on color video images. However, we encounter more and more color video images on day-to-day applications. Thus it's necessary and meaningful to research it.

There're many color image representations [4]. For the convenience, to display colors on the monitor, a true color image is most commonly represented by 24 bits per pixel in a RGB color space with each component R, G and B assigned 8 bits [4]. Therefore, a straightforward way to compress a color image is to split the RGB color image into three channels: red, green and blue, and compress them separately by treating each color component as a single gray-level image. A common alternative to the RGB representation is the YIQ representation, which is the standard used for television transmission. Here Y is the luminance or brightness, I is the hue, and Q is the saturation or depth of color. Luminance refers to color intensity. Hue is the dominant color such as orange, red or yellow and saturation is the amount of white light mixed with a hue. Q and I together are sometimes referred to as chrominance values. Television transmission uses less bandwidth for the hue and saturation components than it does for luminance. Similarly, when compressing color images, the two-chrominance values, Q, and I can be shrunk by between one-quarter and one-half without perceptible image degradation [5].

In this paper we give two kinds of fractal color video compression methods. One method is to split a color video with RGB mode into three channel components, and combine 3 channel components to a long sequence, which is then compressed by fractal gray video compression method. And the other method is compressing a color video in YIQ mode. Firstly, a color video with RGB mode is converted to a color video with YIQ mode. And component Y, I, Q is compressed by a fractal gray video compression method respectively. Experimental results show that the second method can get a higher compression ratio though it has some more computing.

2.FRACTAL GRAY-LEVEL VIDEO COMPRESSION

Let $S = \{ f_t, 1 \le t \le s \}$ be a large sequence of images in a continuous motion, where f_t is a single frame of image, which can be divided into groups of frames (GOF), i.e. $S = \{(GOF)_g : 0 \le g \le l\}$. Each GOF can be viewed as a cubic, which may then be compressed and decompressed as an entity. The principle of fractal video compression method is to compute compression codes according to the self-similarity of an object. In other words, for the every part of an GOF of the sequence S can always be shown similar to another part in the GOF. Hence GOF may be partitioned into non-overlapping small cubes R, known as range cubes. Similarly, each range cube is associated with larger cubes $\tilde{D}$, known as domain cube. R and $\tilde{D}$ depicted in Fig 1.

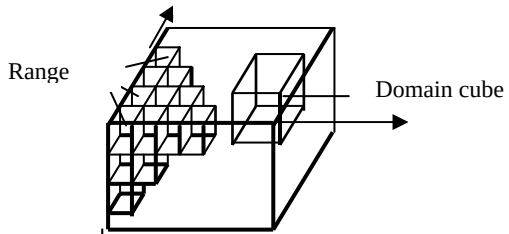


Fig. 1. An indication of range cubes and a domain cube in a GOF

According to the rates of image motion, in general, the size of a range cube, denoted as $n \times m \times l$, whereas the size of a domain cube, denoted as $2n \times 2m \times l$, here n and m could be chosen as 16,8,or 4,and l chosen as 4,3,2 or 1.Thus, a domain cube $\tilde{D}$ is shrunk by averaging the intensities of disjoint groups of four neighboring pixel intensities leading to the same size of range cube R denoted symbolically as D,which is known as a codebook cube.

For every range cube R and codebook cube D there are associated vectors V_R and V_D respectively, defined as $V_R = (r_1, r_2, ..., r_k)$ and $V_D = (d_1, d_2 ... d_k)$, where $K = n \times m \times l$, and $r_i, d_i, i = 1,2,...K$ are the intensities of pixels of R and D. Let $E(D,R)$ be the error function, measuring the difference between R and D, defined as the minimization problem: $E(D,R) = \min_{\alpha,\beta} \| V_R - (\alpha V_D + \beta I) \|$, here α and β are known as scaling and offset factors, and I is the identity matrix of $n \times m \times l$-dimension space.

For a given range cube R, all possible codebook cubes need to be compared in order to find an optimal approximation. In other words, one needs to find a codebook cube D_R satisfies $D_R = \min_D E(D,R)$.

Their some fractal gray-scale video compression algorithms, cube-based compression algorithm, frame-based compression algorithm and hybrid compression algorithm, etc. [6]

3. FRACTAL COLOR VIDEO COMPRESSION

In image processing, there're some typical color spaces including RGB, YIQ and HIS, etc. In this paper we present two new fractal color video compression algorithms, one in RGB color space and the other in YIQ color space.

3.1. Fractal Color Video Compression in RGB Color Space

Let $GOF = \{ f_i, i = 1,2,3,...n \}$ be a sequence of color images with RGB mode in a continuous motion, where f_i is a single frame of image of size $2^M \times 2^N$, $M, N \geq 0$. As we know, a true color image is represented by 24 bits per pixel in a RGB color space with each component, red, green and blue assigned 8 bits respectively. Hence, every color image of GOF can be split into three channels, R, G and B, denoted symbolically as f_{ir}, f_{ig} and f_{ib} respectively.

The main steps of the fractal color video compression algorithm performed in the RGB color space is presented below.

Step 1: Split the sequence of RGB color images GOF into three sequences of gray-level images, $RS = \{ f_{ir}, i = 1,2,3,...n \}$, $GS = \{ f_{ig}, i = 1,2,3,...n \}$ and $BS = \{ f_{ib}, i = 1,2,3,...n \}$;

Step 2: combine RS, GS and BS to a long sequence $MS = RS \cup GS \cup BS$;

Step 3: Apply gray-level fractal video compression algorithm to MS, then get a compression file, $CFile$.

The main steps of the fractal color video decompression algorithm as below.

Step 1: Decompress the compression file, $CFile$ using the corresponding gray-scale fractal decompression algorithm, then get a sequence of gray-scale images, $DM = \{ f'_{1r}, f'_{2r}, ... f'_{nr}, f'_{1g}, f'_{2g}, ... f'_{ng}, f'_{1b}, f'_{2b}, ..., f'_{nb} \}$;

Step 2: Divide DM into three sequences of images, $RS' = \{ f'_{ir}, i = 1,2,3,...n \}$, $GS' = \{ f'_{ig}, i = 1,2,3,...n \}$, $BS' = \{ f'_{ib}, i = 1,2,3,...n \}$,in terms of the order of R, G, B;

Step 3: Combine there frames f'_{ir}, f'_{ig} and f'_{ib} for $i = 1,2,3...n$ to color frames f'_i for $i = 1,2,3...n$, according to the RGB representation characteristic.

Step 4: Joint all color frames f'_i for $i = 1,2,3...n$, then get decompressed color video sequence $GOF' = \{ f'_i, i = 1,2,3,...n \}$.

3.2. Fractal Color Video Compression in YIQ Color

Space

YIQ representation is the standard used for television transmission, where component Y contains the image luminance information or, in other words, it contains the gray scale information. Components Q and I contain the image color information. One of the important advantages of YIQ color space is that this expression can eliminate the correlation between luminance information, Y and chromatic information, I and Q. Moreover, the human visual system is not particularly sensitive to hue and saturation these two color information. We can take advantage of this insensitivity and the independent property in the YIQ color space to shrink Q chromatic values, without perceptible image degradation and I and reduce the compression time.

The calculations performed in the color space conversion from RGB to YIQ are presented below:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.27 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad <1>$$

And an RGB representation can be obtained from YIQ representation by inverting the above transformation as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.00 & 0.96 & 0.62 \\ 1.00 & -0.27 & -0.65 \\ 1.00 & -1.10 & 1.70 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad <2>$$

The compression method is stated as follows:

First, convert a sequence of RGB video, $GOF = \{ f_i, i = 1,2,3,...n \}$, to the YIQ color space using the formula <1>, then get three sequences of images corresponding with Y, I, Q channels, $YS = \{ f_{jY}, j = 1,2,3,...n \}$, $IS = \{ f_{jI}, j = 1,2,3,...n \}$ and $QS = \{ f_{iQ}, i = 1,2,3,...n \}$;

Second, after investigating the distributing of values of I and Q components, it's not difficult to find that the value range of them is concentrative with little changes. Hence I and Q can be assigned with less storage space. $\forall f_{jI} \in IS$ of size $2^M \times 2^N, M \geq 0, N \geq 0$, $u(x, y)$ is the intensity of the pixel (x, y) of the frame f_{jI}. Define the matrix $\mathbf{P}_I$ formed by the pixel intensities of f_{jI} collocated as

$$\mathbf{P}_I = \begin{bmatrix} u(0,0) & u(0,1) & \Lambda & u(0,2^N - 1) \\ u(1,0) & u(1,1) & \Lambda & u(1,2^N - 1) \\ \Lambda & \Lambda & \Lambda & \Lambda \\ u(2^M - 1,0) & u(2^M - 1,1), & \Lambda & u(2^M - 1,2^N - 1) \end{bmatrix}$$

Using less storage space means that rows and columns of the matrix $\mathbf{P}_I$ can be decimated by one-half or more, then get a pixel intensities matrix of a new frame of image. Take decimating by one-half for an example, f_{jI} is shrunk by averaging the intensities of neighboring pixels of disjoint groups leading to a new frame f'_{jI} of size $2^{M-1} \times 2^{N-1}$. The intensity of the pixel (x, y) can be computed as below:

$u'(x,y) = [u(2x,2y) + u(2x+1,2y) + u(2x,2y+1), u(2x+1,2y+1)]/4$,

Where $0 \leq x \leq 2^{M-1}$, $0 \leq y \leq 2^{N-1}$. Then new frames f'_{jI} with $j = 1,2,...,n$ make up of a new sequence of images of I component, $IS' = \{ f'_{jI}, j = 1,2,\Lambda, n \}$. Likewise,

A new sequence of images of Q component $QS' = \{ f'_{jQ}, j = 1,2,\Lambda, n \}$ from QS can be constructed similarly.

Third, apply gray-scale fractal video compression algorithm to YS, IS' and QS', get three compression files, $YFile$, $IFile$, $QFile$.

The main process for decompression consists of the following steps:

Step 1: Apply the corresponding gray-scale fractal decompression algorithm to $YFile$, $IFile$ and $QFile$, then get three sequences of images, $DYS' = \{ df'_{jY}, j = 1,\Lambda, n \}$, $DIS' = \{ df'_{jI}, j = 1,2,\Lambda, n \}$, $DQS' = \{ df'_{jQ}, j = 1,2,\Lambda, n \}$;

Step 2: Reconstruct three new sequences of images $DYS = \{ df_{jY}, j = 1,\Lambda, n \}$, $DIS = \{ df_{jI}, j = 1,2,\Lambda, n \}$, $DQS = \{ df_{jQ}, j = 1,2,\Lambda, n \}$. The reconstruct method is described as follows:

a. $df_{jY} = df'_{jY}, j = 1,\Lambda, n$

b. For df_{jI}' of size $2^{M-1} \times 2^{N-1}$ for $j = 1,2,...,n$, let $w(x, y)$ is the intensity of the pixel (x, y) of df_{jI}'. The matrix

$$\mathbf{P}_{DI} = \begin{bmatrix} w(0,0) & w(0,1) & \Lambda & w(0,2^{N-1} - 1) \\ w(1,0) & w(1,1) & \Lambda & w(1,2^{N-1} - 1) \\ \Lambda & \Lambda & \Lambda & \Lambda \\ w(2^{M-1} - 1,0) & w(2^{M-1} - 1,1), & \Lambda & w(2^{M-1} - 1,2^{N-1} - 1) \end{bmatrix}$$

is formed by pixel intensities of df_{jI}'. Construct image df_{jI} of size $2^M \times 2^N$ using df_{jI}' for $j = 1,2,...,n$. suppose $v(x, y)$ is the intensity of the pixel (x, y) of df_{jI}, the matrix

$$\mathbf{P}_{DI}=\begin{bmatrix} v(0,0) & v(0,1) & \Lambda & v(0,2^N-1) \\ v(1,0) & v(1,1) & \Lambda & v(1,2^N-1) \\ \Lambda & \Lambda & \Lambda & \Lambda \\ v(2^M-1,0) & v(2^M-1,1), & \Lambda & v(2^M-1,2^N-1) \end{bmatrix}$$

is formed by pixel intensities of df_{jI}. $v(x,y)$ can be computed as below:

$$\begin{cases} v(2^x-1,2^y-1)=w(0,0) \\ v(2^x-1,2^y)=w(0,0) \\ v(2^x,2^y-1)=w(0,0) \\ v(2^x,2^y)=w(0,0) \end{cases} (x=0,y=0),$$

$$\begin{cases} v(2^x-1,2^y)=w(0,y) \\ v(2^x-1,2^y+1)=w(0,y) \\ v(2^x,2^y)=w(0,y) \\ v(2^x,2^y+1)=w(0,y) \end{cases} (x=0,0<y\le 2^{N-1}),$$

$$\begin{cases} v(2^x,2^y-1)=w(x,0) \\ v(2^x,2^y)=w(0,y) \\ v(2^x+1,2^y-1)=w(0,y) \\ v(2^x+1,2^y)=w(0,y) \end{cases} (0<x\le 2^{M-1},y=0),$$

$$\begin{cases} v(2^x,2^y)=w(x,y) \\ v(2^x,2^y+1)=w(x,y) \\ v(2^x+1,2^y)=w(x,y) \\ v(2^x+1,2^y+1)=w(x,y) \end{cases} (\begin{matrix}0<x\le 2^{M-1}, \\ 0<y\le 2^{N-1}\end{matrix}).$$

c. Construct a new sequence of images df_{jQ} for $j=1,2,...,n$ from df'_{jQ} is exactly like step b.

Step 3: Combine df_{jY}, df_{jI} and df_{jQ} to a single color frame df_j in YIQ mode, for $j=1,2,...,n$.

Step 4: Use the inverse transformation above to obtain the color frames df_j^{RGB} in RGB mode, for $j=1,2,...,n$;

Step 5: All frames of images df_j^{RGB} for $j=1,2,...,n$, consist of the sequence of decompressed color video, $DGOF=\{df_j^{RGB}, j=1,2,...,n\}$.

4. RESULTS

There are two sequences of motion images in the experiments, one is a sequence of motion images from an extract of a movie consists of frames of 24-bit color image each of 320×208 pixels. The other sequence of motion images is from an extract of a movie consists of frames of 24-bit color image each of 640×320 pixels. The values of compression ratio [6] and PSNR [6] of two compression algorithms are listed in the Table 1.

Table 1. Compression ratios and PSNR obtained from two methods.

Video	Algorithm	Partition	Compression ratio	PSNR
movie 1	1	4×4×3	10.17	32.90
movie 1	2	4×4×3	20.35	32.45
movie 1	1	8×8×1	13.49	29.81
movie 1	2	8×8×1	27.02	29.09
movie 1	1	8×8×2	26.95	29.09
movie 1	2	8×8×2	54.17	28.58
movie 1	1	8×8×3	40.43	28.55
movie 1	2	8×8×3	81.25	28.30
movie 2	1	4×4×3	10.11	29.44
movie 2	2	4×4×3	20.47	28.95
movie 2	1	8×8×1	13.52	27.88
movie 2	2	8×8×1	26.95	26.76
movie 2	1	8×8×2	26.95	27.64
movie 2	2	8×8×2	53.92	26.58
movie 2	1	8×8×3	40.42	26.84
movie 2	2	8×8×3	80.91	26.44

Fig. 2. The first frame of movie 1

Fig. 3. The first frame of the decompressed movie 1 using algorithm 1 partition 4×4×3 (psnr =34.75)

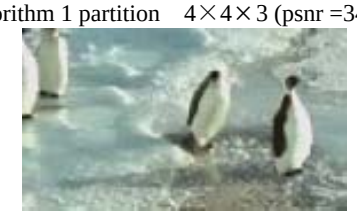


Fig. 4. The first frame of the decompressed movie 1 using algorithm 2 partition 4×4×3 (psnr=33.97)

Fig. 5. The first frame of movie 2

Fig. 6. The first frame of the decompressed movie 2 using algorithm 1 partition 4×4×3 (psnr = 30.44)

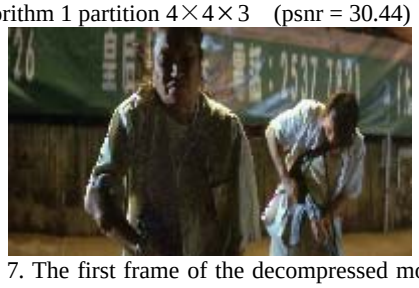


Fig. 7. The first frame of the decompressed movie 2 using algorithm 2 partition 4×4×3 (psnr = 29.00)

5. CONCLUSION

In this paper we propose two new watermarking algorithms based on fractal color video compression method. From the numerical results, we can find that the first algorithm can get a high decompression quality but relative low compression ratio; and the second one get a higher compression ratio without obvious visual differences of the decompressed sequence compared to the first one. At the same time, it should be noted that the fractal color video compression methods possess highly computational complexity. Thus, our future work will combine color transfer [7] with fractal compression method to get a higher compression ratio and decompression quality with less compression time.

6. ACKNOWLEDGEMENT

Meiah Entertainment Ltd, Hong Kong, China, provided the motion images used in the second set of the numerical experiments.

7. REFERENCES

[1] R. Koenen, “Overview of the MPEG-4 Standard – Coding of Moving Pictures and Audio”, Report ISO/IEC JTC1/SC29/WG11 N4668, 2000.

[2] D. T. Lee, “JPEG2000 requirements and profiles – Coding of Still Pictures”, Progress report ISO/IEC JTC1/SC29/WG1 N1803, Hewlett Packard, 2000.

[3] K. Rijkse, Draft ITU-T Recommendation H.263, international Telecommunication Union, 1996.

[4] Gonzalez. & Woods R.E., “Digitial Image Processing”, Reading, MA, Addison-Wesley, 1992.

[5] Y. Fisher (Ed.), “Fractal Image Compression”, Theory and Application, Springer, New York, NY, 1995.

[6] M.Q. Wang, C.H. Lai, “A Hybrid Fractal Video Compression Method”, Computers and Mathematics with Applications 2005, 50:611-621.

[7] E. Reinhard, M. Ashikhmin, et al, “Color transfer between images”, IEEE Computer Graphics and Applications, 2001, 21:34-41.

A Fast Digital Image Transmission approach in the Distributed Maritime Training System *

Dan Liu
Institute of Nautical Science and Technology, Dalian Maritime University
Dalian, Liaoning 116026, P.R.China
Email: dliu_dlmu@126.com
Or
School of Computing, University of Leeds,
Leeds, LS2 9JT, UK
Email: danliu@comp.leeds.ac.uk

ABSTRACT

In the Distributed Maritime Training system, the real-time transmission of large-scale 3D scene is one of the main problems. In recent years Fractal image compression techniques have gained more interest because of their capability to achieve high compression ratios while maintaining very good quality of the reconstructed image. Since Fractal image compression is computationally very expensive, some researchers tried to parallelize the encoding algorithm. Spiral Architecture is a relatively new and powerful approach to general-purpose machine vision system. On this novel architecture, Spiral Addition and Multiplication achieve image processing. In this paper a novel implement based on Spiral Architecture will be presented, keeping the compression ratio of Fractal image compression at maximal to fast the speed of image transmission in the Distributed Maritime Training System

Keywords: Distributed Maritime Training System, Digital Image Transmission, Large-scale 3D Scene, Fractal Image Compression, Spiral Architecture.

1. INTRODUCTION

With the fast development of WEB techniques, it affects profoundly the computer techniques, distributed processing techniques and database techniques. The advantages of WEB are that it's enormous potential of distributed processing, the ability of world-wide information sharing and alternation and abundant data resources; it brings remarkable transforms in the fields of business, industry, science and engineering, as well as the transform in the field of simulation.

Simulation technique is a tool and main part of almost all visual Distributed Training System, E.g. Distributed Maritime Training System (DMTS), such a system achieves realistic training environment utilizing the desktops or laptops through internet or intranet. The aim of it is to provide a high-quality visual training environment, makes the students carry out some training subjects online via browsers everywhere at anytime, and satisfies more users and different demands of learning and training. The advantages of these kinds of systems are the training openness, removability, high mutuality, high integration, easy maintenance, reusage, and saving the cost of exploitation, maintenance and usage, it makes up effectively the disadvantages of the traditional long-distance education, such as localizing listening, reading and watching the preconcerted program, being lack of the chance of practice and experience, less understanding and low retentivity of knowledge. The research contents of these kinds of systems include system designing, architecture, modeling, realization, and the real-time transmission & rendering of large-scale scene. One of the main problems is the real-time transmission of large-scale 3D scene. It is time for us to find out an effective large-scale image compression method.

In recent years Fractal image compression techniques have gained more interest because of their capability to achieve high compression ratios while maintaining very good quality of the reconstructed image. Another intrinsic advantage offered by the Fractal image method is a fast decoding time. Fractal image compression exploits natural affine redundancy present in typical images to achieve a high compression ratio in a lossy compression format. However, Fractal based compression algorithms have high computational demands. It is to say that the main drawback of such techniques is the very high computing time needed to determine the compressed code. To obtain faster compression or feasible compression times for very large images, a novel fast Fractal image compression algorithm is presented.

2. FRACTAL IMAGE COMPRESSION

In the fractal image compression scheme, an arbitrary grayscale image, *T*, of size $I \times I$ is partitioned into two basic block units: the range blocks and the domain blocks.

The range blocks *R* are a set of no overlapping image blocks of size $k = n \times n$, which are denoted as $\{R_i\}_{i=1}^{N_R}$. The number of range blocks is $N_R = \frac{I}{n} \times \frac{I}{n}$, and image *T* is a union of $\{R_i\}_{i=1}^{N_R}$,

$$T = \bigcup_{i=1}^{N_R} R_i \quad (1)$$

* This paper is supported by Fok Ying-tong Education Foundation.(91077).

Overlapping image blocks of *T* with size larger than that of the range blocks are called domain blocks. These domain blocks can be obtained by sliding a window of size $l = m \times m$, where $m > n$, throughout the image to construct the domain pool.

To encode a range block *R*, each of the blocks in the domain pool is scaled to the size of the range block, and is then compared to *R* with respect to intensity offset and contrast parameters, as well as the isometric transformations. The set of contracted domain blocks is denoted as $\{D_i\}_{i=1}^{N_D}$, where N_D is the number of domain blocks in the domain pool. We let $N_D = N_R = n$ throughout this paper.

Then we seek such a contractive operator (a set of Affine transformations) that

$$T = W(T) = \bigcup_{n=1}^{n} \omega_i(D_i) \tag{2}$$

To guarantee the z-contractility of $\omega_1, \cdots, \omega_n$ [1], in case of grayscale images we can use transformations of the form,

$$\omega_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ o_i \end{pmatrix}$$

,where s_i controls the contrast and o_i controls the brightness of the transformation[2,3,4].

The corresponding parameters for the affine transformation ω_i are determined by minimizing the following equation:

$$R_{MSE} = \left\| R - s(D + oI) \right\| \tag{3}$$

Where *I* denotes a unity vector of dimension *k*, and *s* and *o* are the above contrast and offset parameters, respectively, must be determined in advance to calculate the distance between *D* and *R*. The contrast factor should be to ensure the contractility of the transformation. The metric $\|\cdot\|$ is the mean square error (MSE) metric.

3. SPIRAL ARCHITECTURE AND SHM'S OPERATIONS

In this paper the authors introduce a novel Fractal Image Compression method based on a different image structure, namely Spiral Architecture (SA) [5,6,7,8]. This hexagonal image structure was proposed by Sheridan in 1996, on which each hexagonal pixel is identified by a designated positive seven-based integer. These numbered hexagons tile the plane in a recursive modular manner along the spiral direction (see Fig. 1). Based on the spiral addressing system, there are two operations defined on SA, i.e. spiral addition and spiral multiplication.

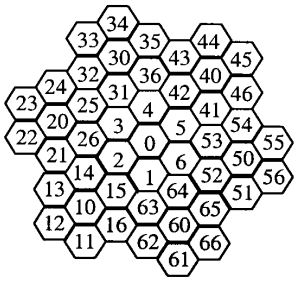


Fig. 1 49 hexagonal pixels labeled by spiral

The definition of these two spiral operations and their applications in our proposed method are briefly introduced in this section. Spiral Counting [9] can be considered as a Spiral movement from a hexagonal pixel to another hexagonal pixel. Any hexagonal pixel in an image can be reached by Spiral counting from any other given pixel in the same image. Spiral counting is used to define two operations in the SA, which are Spiral Addition and Spiral Multiplication. Let *a*and *b* be Spiral addresses of two arbitrarily chosen hexagons in SA.

Definition 1. Spiral addition of *a* and *b*, denoted by $a \oplus b$, is the spiral address of the hexagon found by Spiral counting *b* hexagons in the key of Spiral address 1 from the hexagon with Spiral address *a*.

For example, $3 \oplus 2 = 26$.

Definition 2. Spiral multiplication of *a* and *b*, denoted by $a \otimes b$, is the Spiral address of the hexagon found by Spiral counting *b* hexagons in the key of Spiral address *a* from the hexagon with Spiral address 0.

For example, $15 \otimes 2 = 26$.

It is important to note that each of these two operations takes two spiral addresses and maps them to another spiral address. They can then be used to define two transformations on the spiral address space: image translation (by spiral addition) and rotating image partitioning (by spiral multiplication).

4. FAST FRACTAL IMAGE CODING

Fractal image coding is built on the basis of fractal geometry, its feature is the high compression ratio, fast decoding speed, but coding speed is difficult to reach real time processing, in order to solve the problem, at present, the research on combining fractals and other methods is developed.

Use Spiral operations to transform the original image into 7 small similar images (using Spiral Multiplication to achieve image rotation without scaling, so that the 7 small images have the same direction as the original one). These images are then seemed as domain blocks. Once all transformations have been determined, the transformation information is stored.

Three steps are developed to realize fractal image compression with fast coding speed on Spiral Architecture.

Step1: Segment Operation

Using Spiral operations to transform the original image into 7 small similar images, sorting them as Figure 2;

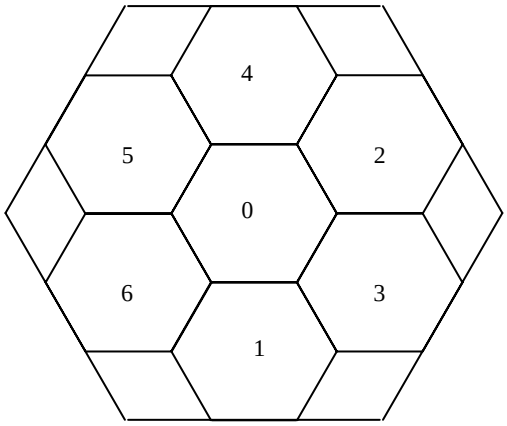


Fig. 2 SHM with size of 7

Step2: Computation Operation

Using Spiral Multiplication to achieve image rotation without scaling so that the 7 small images have the same direction as the original one, computing the following values

$$g_i = \sum_{j=1}^{n} D_j^{(i)} R_j^{(i)}$$

$$h_i = \sum_{j=1}^{n} D_j^{(i+1)}$$

$$k_i = \sum_{j=1}^{n} R_j^{(i)}$$

$$l_i = \sum_{j=1}^{n} D_j^{(i+1)^2}$$

then computing

$$s_i = \frac{n g_i - h_i k_i}{n l_i k_i^{\,2}} \qquad o_i = \frac{k_i - s_i h_i}{n}$$

i=0,1,2,3,4,5,6

Step3: Coding Operation

Using the following 7 Affine transformations to code the image

$$\omega_0 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & s_0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \frac{2}{3} \\ \frac{\sqrt{3}}{3} \\ o_0 \end{pmatrix}$$

$$\omega_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & s_1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \frac{1}{3} \\ 0 \\ o_1 \end{pmatrix}$$

$$\omega_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & s_2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \frac{1}{6} \\ \frac{\sqrt{3}}{6} \\ o_2 \end{pmatrix}$$

$$\omega_3 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & s_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \frac{1}{6} \\ \frac{\sqrt{3}}{2} \\ o_3 \end{pmatrix}$$

$$\omega_4 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & s_4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \frac{2}{3} \\ \frac{2\sqrt{3}}{3} \\ o_4 \end{pmatrix}$$

$$\omega_5 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & s_5 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \frac{7}{6} \\ \frac{\sqrt{3}}{2} \\ o_5 \end{pmatrix}$$

$$\omega_6 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & s_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \frac{7}{6} \\ \frac{\sqrt{3}}{6} \\ o_6 \end{pmatrix}$$

At the decoding phase, the transformation parameters are recursively applied to an arbitrary initial image, and then use the counter Spiral operations to get the decoding image.

5. CONCLUSIONS

We tested this algorithm in a Distributed Maritime Training System (DMTS), which has three tiers Distributed learning system architecture (see Figure 3), and with 100 clients.

The Lena image is used as a test case to analyze runtime, the justification for using the Lena image is its status as a reference image in the image compression literature.

One existing parallel approach to Fractal compression developed employ Quadtree Decomposition [10], we call it QD; we call our method SAP. QD and SAP runtime data for additional MSE values for the Lena image is given in Table 1.

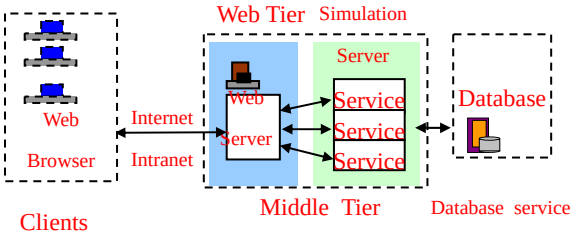


Fig. 3 Three tiers Distributed learning system architecture

Table 1 QD and SAP parallel runtime

MSE		16	32	44
3	QD	169.46	84.32	41.32
	SAP	69.46	30.79	20.03
10	QD	121.93	61.74	30.32
	SAP	46.97	23.02	14.50
20	QD	98.27	49.82	19.09
	SAP	39.58	20.49	12.10
60	QD	68.83	36.01	18.99
	SAP	30.07	16.79	9.74

The hexagonal structure introduces additional complexity to the process of determining which cell a pixel belongs to.

6. REFERENCES

[1] Z.P.Bodó, "Parallel Fractal Image Compression", Master Thesis, Babes-Bolyai University, Cluj-Napoca, p12-13, 2004.

[2] Y.Fisher(ed.), "Fractal Image Compression-Theory and Application", Springer-Verlag, New York, p11-51,1996.

[3] Davoine F., Svensson J., Chassery J. M., "A mixed triangular and quadrilateral partition for Fractal image coding", Proceedings of the IEEE International Conference on Image Processing, Washington D.C, p284-287,1995.

[4] Davoine F., Antonini M., Chassery J. M., Barlaud M., "Fractal image compression based on delauney triangulation and vector quantization", IEEE Transactions on Image Processing, p338-346, 1996.

[5] P.Sheridan,T.Hintz andW.Moore, "Spiral Architecture in Machine Vision" Proc.Australian Occam and Transputer conference,Amesterdam,1991.

[6] E.Schwartz,"Computational anatomy and Functional Architecture of Striate Cortex: spatial Mapping Approach to Perceptual Coding", Vision Research 20, P645-669, 1980.

[7] P.Sheridan,T,Hintz and D.Alexander,"Pseudo-invariant Image Transformations on a Hexagonl Lattice" Image and Vision Computing, Vol.18,p907-917,2000.

[8] P.Sheridan,T,Hintz,"Primitive Image Transformations on a Hexagonal Lattice" Charles Sturt University, Bathurst, NSW, Australia, Tech. Report, December 2000.

[9] Qiang Wu, Xiangjian He and Tom Hintz, "Image Rotation without Scaling on Spiral Architecture", Journal of WSCG, Vol.10, No.2, Vaclav Skala(ed.), p515-520,2002.

[10] David J.J. and Wagdy M., "Parallel Pipelined Fractal Image Compression using Quadtree Recompression", The Computer Journal, Vol.39 (1), p10-23, 1996.

Target Detection Algorithm of Underwater Sonar Image

Xiaodong Tian，Zhong Liu，Dechao Zhou
Department of Electronic Engineering, Naval University of Engineering
Wuhan, Hubei Province,.China
Email: txdqtxy@163.com

ABSTRACT

Aiming at terms of target detection in high-frequency sonar image. Firstly, target detection algorithm based on fractal features was depicted; secondly, based on analysis of pixel's grey-level distribution function in the sonar image, a novel algorithm used to target detection was presented. In the regions of sliding window around each pixel in sonar image, this method estimated distribution function parameters and error between grey-level distribution function and normalized histogram as features, which were adopted to construct characteristic images of sonar image. Then adaptive threshold arithmetic was used to detect target fast and exactly. Simulation results indicate that this method has characteristics of real-time and high-precise. Besides, it can overcome the influence of natural objects such as rock to realize extract man-made target. Then, this algorithm can improve capability of underwater detection effectively.

Keywords: Target detection, Gray-distribution model, Fractal theory, Man-made target, Sonar image.

1. INTRODUCTION

In recent years, underwater detection technology has been developed greatly and applied in many fields such as military detection, sea-bottom measurement and pipe-inspection, etc. imaging sonar is one of most important instruments of underwater detection. Especially, synthesized aperture sonar (SAS) improves detection range and resolution with low frequency.

Due to complexity of underwater acoustic environment and nonlinear imaging of sonar instrument, underwater sonar image has characteristics such as low-contrast and poor quality, which brings difficulties to man-made target detection. The fractal-based method is one of most common methods for man-made target detection. At present, this method has been applied in many fields. Simultaneously, experiments indicate that pixels grey-level value of sonar image submits to certain probability distribution model. This model can used to image filter and segmentation. Generally the grey-level distribution model is adopted as prior acknowledge for Random fields segmentation algorithm and CFAR detection in existing literatures. And in this paper, the grey-level distribution model of reverberation background region in sonar image was simulated and analyzed. Based on these results, a novel target detection method was presented. In this algorithm, parameters and matching error of distribution model were chosen as features.

2. FRACTAL-BASED TARGET DETECTION ALGORITHM

The most physical basement of this method is the difference in surface coarseness term between man-made target and natural scenes. Man-made target usually has simple surface texture and regulated geometric shape, so its fractal dimension is less than natural scenes in fractal model. Reference [1] adopted fractal model to target detection of sonar image and integrated with matching filter, detection precision and robustness was improved.

Many methods have been presented to estimate fractal parameters, such as blanket bestrow method, box-counting method, power spectrum method, etc. estimated result of every method is different. At the same time, the definition of fractal parameter is various. The extended fractal (EF) feature is one of these definitions [2]. It derives from multi-scale Hurst parameter, and can be used to target detection efficiently.

Supposing $I(m,n)$ represents original image, the EF features in x-axis direction and y-axis direction can be defined as:

$$F^x(m,n)=\frac{1}{2}\log_2\left[\frac{f_\Delta^x(m,n)}{f_{2\Delta}^x(m,n)}\right]$$
$$F^y(m,n)=\frac{1}{2}\log_2\left[\frac{f_\Delta^y(m,n)}{f_{2\Delta}^y(m,n)}\right] \quad (1)$$

Where, f^x, f^y is constructing function in x-axis direction and y-axis direction respectively. The size of sliding window is chosen as $(2w+1)\times(2w+1)$, then:

$$f_\Delta^x(m,n)=\sum_{i=-w}^{w}\sum_{j=-w}^{w}\left|I(m+i+\Delta,n+j)-I(m+i-\Delta,n+j)\right|^2$$

$$f_\Delta^y(m,n)=\sum_{i=-w}^{w}\sum_{j=-w}^{w}\left|I(m+i,n+j+\Delta)-I(m+i,n+j-\Delta)\right|^2 \quad (2)$$

Considering the isotropy of EF feature, the EF feature of centering pixel in sliding window can defined as following:

$$F(m,n)=\frac{F^x(m,n)+F^y(m,n)}{2} \quad (3)$$

3. DETECTION ALGORITHM BASED ON GREY-LEVEL DISTRIBUTION MODEL

3.1 Grey-level distribution model and parameter estimation

In fields of Radar image processing, grey-level distribution model has been studied deeply, the most models include K-law model, Gamma law model[3], etc. But aiming to sonar image, this kind of study is very few, and almost used as prior knowledge for random field segmentation and CFAR target detection. The following several models are common, for example Weibull law [4, 5], Gaussian law, Rayleigh law and K-law[6,7], etc.

1) Rayleigh law model[4]

In fact, Rayleign law is the special form of Weibull law as shape parameter $c = 2$:

$$W_Y(x;\min,2,a) = \frac{2\cdot(x-\min)}{a^2}\cdot\exp\left(-\frac{(x-\min)^2}{a^2}\right) \quad (4)$$

Where, $\min$ represents the minimum value among image pixels grey-level values; x represents grey-level value of each pixel; a is scale parameter.

Then the scale parameter a can be estimated as following:

$$\hat{a} = \left(\frac{1}{M}\sum_{s=1}^{M} x_s^{\hat{c}}\right)^{1/\hat{c}} \quad (5)$$

Where, $\hat{c} = 2$; M represents the total number of pixels.

2) K-law model[8]

The probability distribution function of K law model can be defined as:

$$p(x) = \frac{2}{x\Gamma(v)\Gamma(L)}\left(\frac{Lvx}{\mu}\right)^{\frac{L+v}{2}} K_{v-L}\left(2\sqrt{\frac{Lvx}{\mu}}\right) \quad (6)$$

Where, μ is mean value of referenced unit; v is shape parameter; L represents image view-number; $\Gamma(\cdot)$ is Gamma function; K_{v-L} represents modified Bessel function; in above formulate, integral calculation is very complex, certain approximating method is always adopted.

Mean and shape parameter can be estimated as:

$$\mu = \frac{1}{N}\sum_{i=0}^{N-1} x_i \quad (7)$$

$$\left(1+\frac{1}{v}\right)\left(1+\frac{1}{L}\right) = \frac{\frac{1}{N}\sum_{i=0}^{N-1} x_i^2}{\left(\frac{1}{N}\sum_{i=0}^{N-1} x_i\right)^2} \quad (8)$$

3) Gamma law model[9]

The probability distribution function of Gamma law model can be depicted as following:

$$p(x) = \frac{\beta^v}{\Gamma(v)} x^{v-1}\exp(-\beta x),\ x > 0 \quad (9)$$

Where, v is shape parameter; β is scale parameter; $\Gamma(\cdot)$ is Gamma function.

In this model, shape parameter v and scale parameter β can be estimated as :

$$v = \frac{m_1^2}{m_2 - m_1^2};\quad \beta = \frac{m_1}{m_2 - m_1^2} \quad (10)$$

Where, $m_r = \frac{1}{N}\sum_{i=0}^{N-1} x_i^r$, $r = 1,2$.

4) Matching error

The matching degree between grey-level distribution model and normalized histogram can be evaluated by Kolmogorov distance criterion[7], In this paper, matching error was adopted. The evaluating function can be defined as:

$$dk = \sum_{i=1}^{G}\left|h(i) - p(i)\right| \quad (11)$$

Where, h represents normalized histogram of grey-level image; p is probability value of distribution model; G is image grey-level, and $G = 256$.

3.2 Target detection algorithm

As mentioned above, the existing study of grey-level distribution model in sonar or radar image almost faced to image segmentation, this model can be used as prior knowledge or adopted to CFAR target detection. Then the following problems can be unavoidable:

1) the image segmentation method based on random field model has great calculation burden and poor efficiency;

2) the estimated parameter of distribution model and the matching error information lack sufficient utilization.

Therefore in this paper, based on analysis of grey-level distribution model of sonar image, a novel method for target detection in sonar image was presented. Considering in background region, there is approximated united distribution model for each sub sliding window and matching error is little; but in regions of target edge, pixels grey-level value has great changing, there is not united distribution model for two side regions of target edge. Therefore, model parameters and matching error can be adopted as features to construct feature map. The total steps of this method can be depicted as following:

1) Choosing the size of sliding window. The size of sliding window has influence on efficiency and target positioning precise of this method. The rule of choosing is some factors such as target size in image, estimating precision of distribution model parameters. If the size is too small, the number of pixels in sliding window is little, then estimating error of model parameters is great; on the contrary, if the size is too big, for small target, occupied pixels of target is few, then the contrast between target region and background region. Based on many experiments and simulations, the size of sliding window can be chosen in section $[11,21]$, in this paper the size 15×15 is selected.

2) From top to bottom, and from left to right, sliding window moves in original image. At each pixel point, the distribution model parameters and matching error are calculated;

3) Constructing feature maps based on parameters matrix and matching error matrix;

4) Self-adaptive thresholding method is applied to segment feature map, and morphological operators is used to filter little noise disturbance. The whole flow chart is illustrated as Fig. 1.

Limited to paper length, Gamma law model and Rayleigh law model were taken for target detection as examples in this paper. Fig. 2. (a) shows original sonar image. This sonar image comes from Klein 5000 engineering software. There are five man-made targets and many natural scenes such as rocks in this image. Fig. 2. (b) and (c) represents scale parameter feature map and matching error feature map based on Rayleigh law model respectively, among these both images feature values in man-made target region has marked distinction with background region, and background region is smoothed. At the same time, Fig. 3.(a)-(c) illuminates shape parameter v feature map, scale parameter β feature map and matching error feature map based on Gamma law model respectively. Self-adaptive thresholding method is

applied and results are depicted as Fig. 4. Fig. 4.(a)-(c) represents original image detecting result, Rayleigh law matching error map detecting result and Gamma law matching error map detecting result respectively. Analyzing these results, we can know that the detection precision of the method presented in this paper is high and its anti-disturbance ability is strong.

Simultaneously, the fractal method mentioned in section 2 was experimented, the detection results based on fractal feature[1] and extended fractal feature[2]are illuminated as Fig.5 and Fig.6. in these results, only four target are detected and one target is leaked. The positioning veracity is not precise and the anti-disturbance performance of EF feature is poor.

According to feature maps, the contrast between man-made target region and background region is great especially in feature map based on matching error.

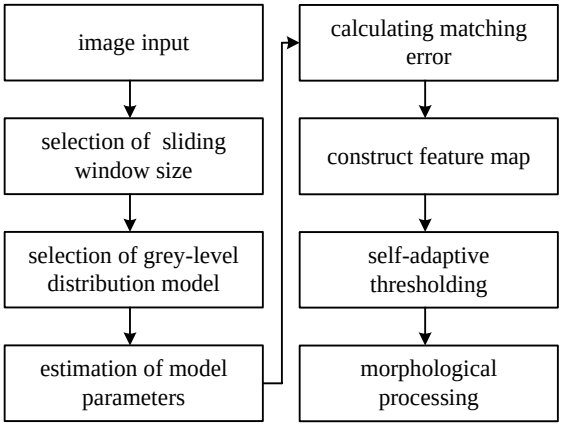
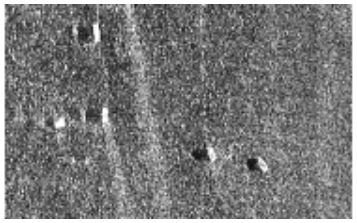


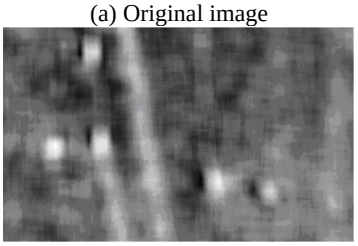
Fig. 1. Flow chart of target detection

From simulation process and results, several conclusions can be derived:

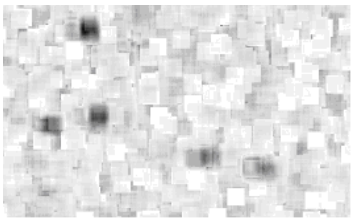
1) The method presented in this paper has more precise target position and robustness than traditional methods such as fractal model. In particular, detection method based on matching error feature, the detection



(a) Original image

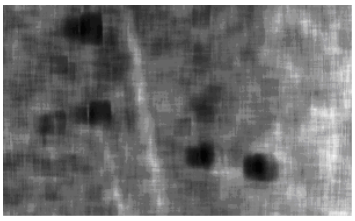


(b) a Parameter

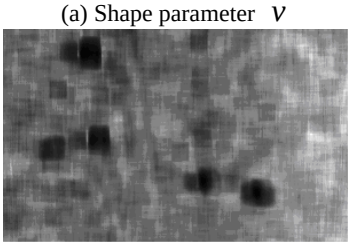


(c) Matching error

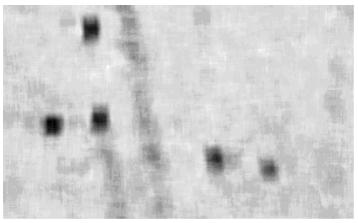
Fig. 2. Original image and Rayleigh law feature map



(a) Shape parameter v

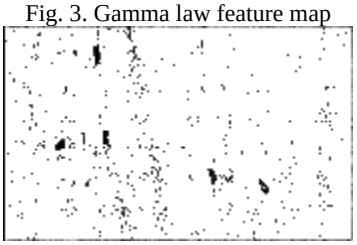


(b) Scale parameter β

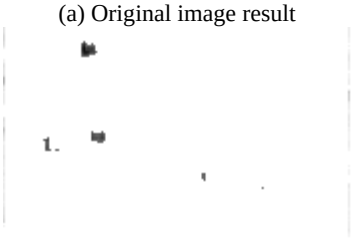


(c) Matching error

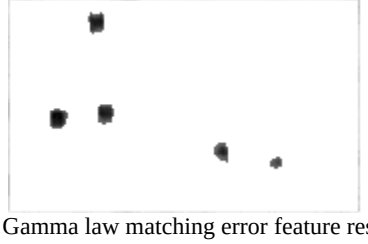
Fig. 3. Gamma law feature map



(a) Original image result



(b) Rayleigh law matching error feature result



(c) Gamma law matching error feature result

Fig. 4. Self-adaptive thresholding result

Fig. 5. Detection result based on fractal feature

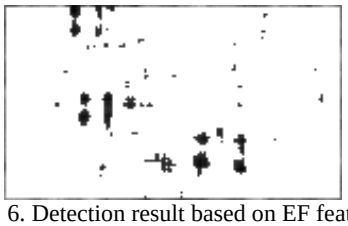


Fig. 6. Detection result based on EF feature

performance is prominent. Viewed from real-time term, fractal method requires longer calculation time, but the method in this paper is more efficient;

2) The selection of grey-level distribution model is very important in this detection algorithm. If improper model is chosen, then the contrast between background region and target region in feature map is poor, which bring difficulty to the following thresholding processing. Moreover, the calculation burden of different distribution models is various;

3) With data fusion theory, detection precision will be improved and false alarming ratio will be decreased by integrating other features such as target shape, target size, etc;

4) To further increase calculation speed, sonar image can be divided into many sub-section images, which are not overlapped or part overlapped. Model parameters and matching error of each sub-section image are calculated. This method will decrease target-positioning precision.

4. CONCLUSION

Based on analysis of pixel's grey-level distribution function in the sonar image, a novel algorithm used to target detection was presented. In the regions of sliding window around each pixel in sonar image, this method estimated distribution function parameters and error between grey-level distribution function and normalized histogram as features, which were adopted to construct characteristic images of sonar image. Then adaptive threshold arithmetic was used to detect target fast and exactly. Simulation results indicate that this method has characteristics of real-time and high-precise. Besides, it can overcome the influence of natural objects such as rock to realize extract man-made target. Through contrary and analysis of simulation results, the selection of the size of sliding window and grey-level distribution model has certain influence on calculation efficiency and parameter estimating precision.

REFERENCES

[1] B.Wang, J.Tian, "Target detection in underwater acoustic images", Journal of detection & control, 2004, Vol.26 (4): 34-37.

[2] W.Yang, H.Sun, "Detection of ships and ship wakes in spacebarne SAR imagery", Geomatics and information science of Wuhan University, 2004,Vol.29 (6): 682-685.

[3] M.J.Lu, R.S.Wang, "Reliable image segmentation based on Markov random field model", ACTA ELECTRONICA SINCA, 1999, Vol.27 (2): 87-89.

[4] M.Mignotte, C.Collet, "Sonar image segmentation using an unsupervised hierarchical MRF model", IEEE TRANSACTION ON IMAGE PROCESSING, 2000, Vol.9 (7): 1216-1231.

[5] M.Mignotte, C.Collet, "Three-class Markovian segmentation of high-resolution sonar image", Computer vision and Image Understanding, 1999, Vol.76 (3): 191-204.

[6] Brian R.La Cour, "Statistical Characterization of active sonar reverberation using extreme value theory", IEEE Journal of oceanic engineering, 2001, Vol.29 (2): 310-316.

[7] Douglas A.Abraham, Anthony P.Lyons, "Reverberation envelope statistics and their dependence on sonar bandwidth and scattering patch size", IEEE Journal of Oceanic Engineering, 2004, Vol.29 (1): 126-137.

[8] Gareth R.Elston, Judith M.Bell, "Pseudospectral time-domain modeling of non-Rayleigh reverberation: synthesis and statistical analysis of a sidescan sonar image of sand ripples", IEEE Journal of Oceanic Engineering, 2004, Vol.29 (2): 317-329

[9] L.Li, F.Deng, "CFAR target detection in SAR imagery", Journal of test and measurement technology, 2002, Vol.16 (1): 9-13.

Visualization of Three-Dimensional Organic Data Based on CT image*

Xiaomin Cheng , Cui Zhi
School of Mechanical Engineering, Ningbo University of Technology, Ningbo, China, 315016,
Email: yzbchxm@nbip.net, chxm369@hotmail.com

ABSTRACT

Discuss how to use the digital image processing technology and realize the visualization of three-dimensional data of tissue and organs. Through internal segmentation and edge detection, the edge extraction and tracking were fulfilled and three-dimensional CT image data was obtained. And based on the characteristics of medical images, the following algorithms were proposed: the minimal diagonal algorithm which solved the big curvature and distortion problem; the dummy point bridge algorithm which solved the branching problem; the transition layer algorithm which solved the holes in skeleton problem. Finally, validate the algorithms through a human ear visualization case, in the same time, three-dimensional data was compressed to solve the data redundancy problem. All these steps ensure the accuracy of visualization and obtaining the satisfying organ images. The visualization result can be further used by CAD and RP system, which produce the prototype of tissue and organs.

Key words: three dimensional data, tissue and organ, visualization.

1. INTRODUCTION

Based on various tissues, the density fields from MRI and CT will correspond to various density values. According to the pathological change regions presented by multiple profiles from various directions or re-construction of the three dimensional real images showing more details, doctors can identify the sizes and positions of diseases not only by quality but also by quantity. Such a data visualization approach plays more important role to the more complex regions such as brain. Based on the proposed real-to-virtual method, doctors can easily locate the disease parts, determine effective surgery plans, and carry on simulation surgeries before taking actual ones [1].

Each CT layer is in fact a two-dimensional data field. The layer gradation represents object density. A large amount of three-dimensional data were obtained after arranging the layers in certain order and being processed such as edge examination and track, edge withdrawing and identification, etc. The obtained data has composed a three-dimensional data field. This study mainly focused on visualization of the obtained data field. The proposed approach was verified by visualization of the ear profile of human being and around tissues as an example.

* The Project Supported by Zhejiang Provincial Natural Science Foundation of China: 502140;
The Project Supported by Ningbo Youth Foundation of China: 2003A62021

2. CT 3-DIMENSIONAL FIELD DATA VISUALIZATION ALGORITHM

In order to obtain third-dimension graph, this study uses light track algorithm to carry on organ visualization. First, the complete object surface database structure was set up. It then was transformed into world coordinate system. After defining plane, screen, and viewpoint of the projection, the data structure was traced by light track and demonstrated. At the same time, the projection and cutting-out were completed.

2.1 Lights tracks algorithm

In the light tracking system, a viewpoint was selected within the object space. A suitable plane rectangle region was selected as the projection screen appropriate to the selected views point [2]. The viewpoint was on the Z-axis, and the XY plane was selected as the projection screen that was divided into several small square grids by two groups of mutually perpendicular parallel lines. Each small square grid corresponds to one picture element on the monitor screen. The sampling point was located at the center of the grid.

The brightness of colors was calculated for each picture element, which then resulted in the whole picture. [3]. Three parts consists the brightness I_λ, which is the radiation brightness source from the V views point from the object surface:

1) The reflected light brightness caused by the directly illustrated from the light source, named as $I_{l\lambda}$. Its value was calculated using local illumination model;

2) The reflected or refracted light brightness caused by the light reflected from other objects along R direction source from direction V of glass, named as $I_{s\lambda}$;

3) The reflected or refracted light brightness coming from V along the T transmitting direction, named as $I_{t\lambda}$.

Thus the overall illumination model equation (1) was obtained as follows:

$$\begin{aligned} I_\lambda &= I_{l\lambda} + K_s C_{s\lambda} I_{s\lambda} + K_t C_{t\lambda} I_{t\lambda} \\ &= K_a C_{d\lambda} I_{a\lambda} + \sum_{i=1}^{m} S_i f(d_i) I_{p_i\lambda} [K_d C_{d\lambda} (L_i \cdot N) \\ &\quad + K_s C_{s\lambda} (H_i \cdot N)^n] + K_s C_{s\lambda} I_{s\lambda} + K_t C_{t\lambda} I_{t\lambda} \end{aligned} \quad (1)$$

in which,

K_t: transmission coefficient of object, values between 0 and 1; K_s: reflection coefficient of object, values between 0 and 1; $C_{t\lambda}$: color coefficient of transmission, values depending on various colors. If $C_{t\lambda}$= 0, then the color light cannot penetrate the object; $C_{s\lambda}$: color coefficient of reflection. If $C_{s\lambda}$= 0, then there is no reflection; K_s, $C_{s\lambda}$ and $I_{s\lambda}$: contribution of the brightness of all reflection lights to picture elements; K_t, $C_{t\lambda}$ and $I_{t\lambda}$: contribution of the brightness of all transmission lights to picture elements.

The light track process without limits is impossible to

work. Therefore, a termination condition was given to Equation (1): because K_s and K_t all are smaller than 1.0. After light reflection and refraction, its brightness will become weak. So a threshold value was set up in advance. During light tracing, if the picture element brightness of the traced light is smaller than the defined threshold value, the tracing will stop.

2.2 Intersect computation

1) Inspection of ambient ball

The main focus of light tracing algorithm is to find the intersection point of light and object surface. In light tracing algorithm, large amount of efforts have to focus on the Intersect computation. In order to increase the efficiency of Intersect computation, ambient ball method is employed to predict if there is intersection between object and light. If intersection exists, Intersect computation will carry out.

The maximum and minimum values of the coordinates at the object peak point are supposed to be x_{max}, x_{min}, y_{max}, y_{min}, z_{max}, z_{min}. Then the coordinates at the center of ambient ball of (x_0, y_0, z_0) are:

$$x_0 = \frac{1}{2}(x_{max} + x_{min}), \quad y_0 = \frac{1}{2}(y_{max} + y_{min}), \quad z_0 = \frac{1}{2}(z_{max} + z_{min}) \tag{2}$$

The radius of the ambient ball is:

$$r = \frac{1}{2}\sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2 + (z_{max} - z_{min})^2} \tag{3}$$

If the distance of d between the random point along light of $O = [x, y, z]^T$ and the center of ambient ball of (x_0, y_0, z_0) is shorter than the distance between any other random point along light and center of ambient ball, then d is selected as the distance between light and ambient ball center, and the straight line with d must be perpendicular to this light. Value of d can be computed by the following equation:

$$d^2 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 \tag{4}$$

If $d_2 > r_2$, then light will not intersect with ambient ball, and also cannot intersect with the object surrounded by the ambient ball. Otherwise, it is necessary to locate the intersection point between light and the surface of object.

2) Intersect computing of light and simple polyhedron

If the object is a simple polyhedron, its surface is composed by plane polygons. Then the problem of Intersect computation of light and object surface becomes the Intersect computation of light and plan polygons. Now the polygon is supposed to be P_0, P_1,⋯, P_n, then its plane equation can be expressed by:

$$N \cdot O + d = 0 \tag{5}$$

The parameter equation of light is as follows:

$$O = Q + tE, \quad t \in [0, +\infty] \tag{6}$$

Based on equations of (6) and (5), the following equation can be obtained:

$$N \cdot (Q + tE) + d = 0 \tag{7}$$

If $N \cdot E = 0$, light is parallel to the plane polygon without intersection. If $N \cdot E \neq 0$, the parameter of the intersection between light and the plane can be expressed by:

$$\bar{t} = -(d + N \cdot Q)/(N \cdot E) \tag{8}$$

If $\bar{t} > 0$, the intersection point $\overline{O}$ between light and plan polygon can be obtained by inputting $\bar{t}$ into Equation (6) (if $\bar{t} \leq 0$, there is no intersection):

$$\overline{O} = Q - \frac{d + N \cdot Q}{N \cdot E} E \tag{9}$$

Now it is necessary to determine if intersection point $\overline{O}$ located within the polygon plan. So both polygon and intersection point $\overline{O}$ are projected onto one plan. If the projection of intersection point $\overline{O}$ is located within the projection of polygon plan, then intersection point $\overline{O}$ should intersect with polygon plan. Or else, there is no intersection between light and polygon and no Intersect computing is necessary.

In Equation (9), N represents the unit vector of plan;

$Q = [x_Q, y_Q, z_Q]^T$ — the starting point of the light;

$E = [x_E, y_E, z_E]^T$ — unit vector and the light direction.

3) Intersect computing of light and quadric

The quadric equation may be expressed by:

$$f(x, y, z) = [x, y, z, 1] \cdot W \cdot [x \quad y \quad z \quad 1]^T = 0 \tag{10}$$

W is the matrix coefficient.

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & j \end{bmatrix}$$

Inputting into Equation (10), then:

$$O^T W_{11} O + O^T W_{12} + W_{21} O + j = 0 \tag{11}$$

Inputting Equation (6) into (11), then:

$$at^2 + bt + c = 0 \tag{12}$$

in which,

$$a = E^T W_{11} E, \quad b = E^T (W_{11} + W_{11}^T) Q + (W_{12}^T + W_{21}) E$$

$$c = Q^T W_{11} Q + (W_{12}^T + W_{21}) Q + j$$

Similarly, the parameters of intersection point can be obtained.

2.3 Normal vector computation

It is possible that various intersection points exist if light intersects with various objects. Based on the comparison of various intersection points, the intersection point most close to the views point can be determined. This intersection point is visible and locates along the direction of the view. In order to do following computation, the normal vector at the visible point on the curve must be determined.

For quadratic surface, the normal vector at point (x, y, z) is the unit of the vectors expressed by the following equation:

$$(\frac{\partial Q}{\partial x},\frac{\partial Q}{\partial y},\frac{\partial Q}{\partial z})=(2Ax+Dy+Fz+G,\ 2By+Dx+Ez+H,2Cz+Ey+Fx+J) \quad (13)$$

For parametric surface, the normal vector at its point of $(x_0,y_0,z_0) = (x(u_0,v_0),y(u_0,v_0), z(u_0,v_0)$ is:

$$(\frac{\partial x}{\partial u},\frac{\partial y}{\partial u},\frac{\partial z}{\partial u})\times(\frac{\partial x}{\partial v},\frac{\partial y}{\partial v},\frac{\partial z}{\partial v})\Big|_{\substack{u=u_0\\v=v_0}} \quad (1\ 4)$$

which is the cross product of two tangent vectors along the direction of parameter line through point (x_0, y_0, z_0).

3. SEVERAL KEY QUESTIONS OF VISUALIZATION OF COMPLEX TISSUES AND ORGANS

Visualization of Medicine image is a comparatively new research field. This field has attracted attention domestically since 1990s. At present, it is still at development phase. It is necessary doing further study. Several key questions during visualization were studied in this research in order to improve the accuracy and precision of visualization of tissues and organs.

3.1 Algorithm improvement to curve with great curvature

Great curvature means one part of curve on the surface is very sharp with high curvature. Big distortion means the shapes based on the upper and lower surfaces have huge difference because of CT level distance or relative slope. Smallest diagonal process method is to connect the surfaces on the two layers based on the similarity between neighboring medicine fault image. Therefore, such kind of curve with great curvature and big distortion is a difficult problem during resulting in triangular surface. Fig.1 presents such kind of problem, which shows a big angle between AB line on outline C_K and CD line on C_{K-1}. So the length of AD line is always smaller that that of BC line, which is out of great curvature range and smallest diagonal process method can only be used after point E. The lines between point B and C_{K-1} can result in a triangular surface, which can make some errors if smallest diagonal process method is employed directly, as shown on Fig.2. So additional constraint condition was included in order to avoid self-intersect, which is to determine the angle curve surface SACO. The triangular with the smallest angle between triangle surface of SABC, SADC and the produced should be what is looking for. Such method presents good result in practical work, as show on Fig.3.

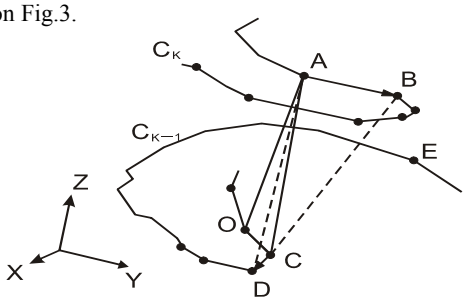


Fig. 1. Great curvature distortion situation

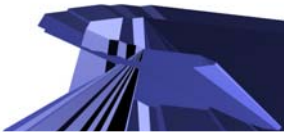


Fig. 2. Great curvatures distortions inbred line wrong

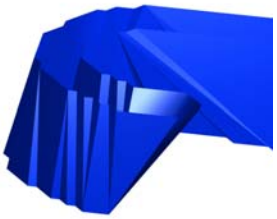


Fig. 3. Great curvatures distort the correct outline

3.2 Branch processing

Visualization of tissues and organ has branch-processing problem. Fig.4 shows the three kinds of theoretically branch forms, respectively are A, B and the C branch. During visualization, firstly the branch was determined belonging to which one shown on Fig.4 based on the descriptions of the shapes of tissues and organs. If it belongs to A or C branch, what need to do is only simply to do triangular distribution and close the end surface. Branch B situation is important. As shown on Fig.5, the cross section of C_{K-1} on S_{K-1} should connect with the other cross sections of C_{K1} and C_{K2} located on S_K, which belong to branch B. In this study, the visualization method is as follows:

Firstly, the two points of T_i, L_j with the shortest distance between two branch section surfaces were located, as shown on Fig.5. Then a virtual point of S located at level $(S_{K-1}+S_K)/2$ were created, which coordinates are:

$$\begin{cases} X_S=(X(Ti)+Y(Lj))/2; \\ Y_S=(Y(Ti)+Y(Lj))/2; \\ Z_S=(S_{K-1}+S_K)/2; \end{cases} \quad (15)$$

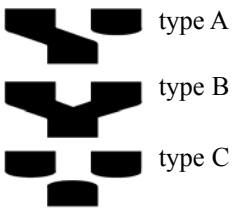


Fig. 4. Theoretically branches situations

The virtual point of S is acting as a factor connecting the two branch surfaces together. After reorganizing the surface points, the two branches became one closure

outline. The data was saved into database and the three-dimensional restructure was completed. Fig.6 presents the visualization of ear surface using the above method, which obtained a good result.

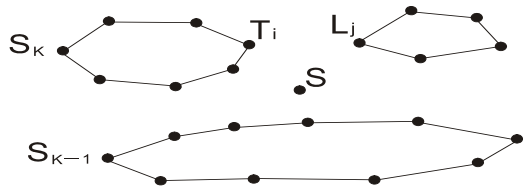


Fig. 5. B branch hypothesized selects S the

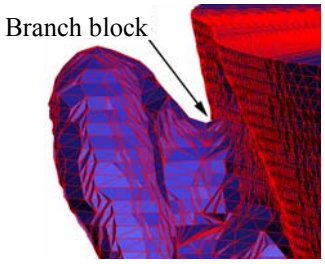


Fig. 6. Correct processing branches situation

3.3 Forming of a figure with a hole

Besides branch situation, because of the complexity of tissues and organs, there maybe exist the visualization problems of figures with holes and solid. Such as a tissue organ with an incomplete hole, if its end surface is not normal to the cutting section, there exists three different kinds of cross section surfaces during fault section scanning with CT, as shown on Fig.7. They were given names as of "C", "O", and "middle" (cutting located exactly on the boundary line) types. Based on its special shape, the "middle" type was cutting at point M between interior figure $M\text{-}P_2\text{-}Q_2\text{-}R_2\text{-}M$ and outer figure $M\text{-}P_1\text{-}Q_1\text{-}R_1\text{-}M$. Or it can be treated as another figure of $M\text{-}P_1\text{-}Q_1\text{-}R_1\text{-}M\text{-}R_2\text{-}Q_2\text{-}P_2\text{-}M$. The "middle" type is difficult to obtain because of its cutting point at exactly at the boundary point. The usual situations are "C" and "O" types during CT fault scanning. It is quite difficult to connect C and the O figures directly, moreover, which cannot reflect the primitive shape of object. It is necessary to create a middle layer between two layers, which can be as a transitional layer. The actual steps are firstly finding the coordinates of point M. In general, the most close two points T_i and L_j on O type figure may correspond the X axis and the Y axis coordinates of M point. The coordinates at point M may be obtained by the following equations:

$$\begin{cases} X_M = (X(T_i) + X(L_j))/2 \\ Y_M = (Y(T_i) + Y(L_j))/2 \\ Z_M = (S_{K-1} + S_K)/2 \end{cases} \quad (16)$$

The X-axis and the Y-axis coordinates of other points of "middle" type can be obtained by copying all points of O type figure. Coordinates of Z point remained the same as those of M point. Then reorganizing the points based on the layers relationship can result in a normal figure.

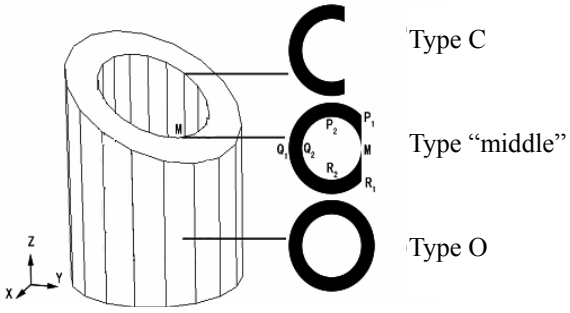


Fig. 7. Cross section with holes

4. VISUALIZATION EXAMPLE

In the following paragraphs, an ear figure of human being was employed as an example demonstrating the visualization of tissue organs. In addition to the above-discussed problems during visualization of tissue and organ, this study was also looking for solving the other two problems of three-dimensional data optimization and the end surface closure of ear.

During visualization of ear, the information of CT image in brain from other parts of only provide invalid information, but that information is essential for the complete closure of the whole outer figure of the ear. Therefore the control points of other parts in addition to ear have to be compressed by weighting factor. The actual realization method is: assigning the redundant region by the user; according to a threshold value, a valid point is selected by computer in a range of some points of the original figure database, and other points were deleted; finally, a new point sequence are created, in which only the information about the approximate outer surface remains in this region.

The threshold value may be determined based on the own situation by the user. Based on this method, a large amount of invalid information points can be deleted based on the previous control point optimization, which can increase the efficiency on the basis of the selected threshold value, as shown in Table 1 to show.

Table 1. optimization comparative table

CT scanning Layer	Original amount of points	Optimization amount of points	Efficiency increased
1st Layer	1557	340	78.2%
23rd Layer	1805	384	78.7%
57th Layer	1816	312	86.9%
63rd Layer	1629	307	81.1%

Because of the specific features of ear and cutting direction of CT fault images, the closure problem of figure may exists during visualization. The figure of an ear is a continual entity, and it could not present the actual situation to close the end surface as a plan surface. Therefore the end surface was assumed to be a cone-shape in this study. The algorithm is to find the

gravity center of O of all points. The coordinates of point S along X-axis and Y-axis may be obtained by the formula for center of gravity, presented in Equation (17) as follows:

$$\begin{cases} O_x = \sum X / N \\ O_y = \sum Y / N \\ O_z = Z_s + 0.5h \end{cases} \quad (17)$$

Then, all points along the edge are connecting to the point O. The Z coordinate of point O is corresponding to all Z coordinates of all other edge points and increases or decreases half CT separating layer gap. As shown on Fig.8, it is obviously better expressing the actual situation based on the above method compared to the method employing a polygon representing the end surface dealing with the end surface of figure. Fig.9 shows the visualization results of ear figure and around areas. It can speed visualization by compressing the data of the parts farther away from the ear, although which also lower the visualization accuracy. From the final visualization results, the visualized figure presents well of the waves of the ear surface and also the three dimensional results were obtained to the ear figure, which also demonstrated that a satisfied visualizing results can be obtained to simulate the surface of an object. The visualization method was also verified in Pro-E and 3DSMAX. The output is in STL format. Such medical model produced in rapid prototyping system can be also used in the medical surgery simulation.

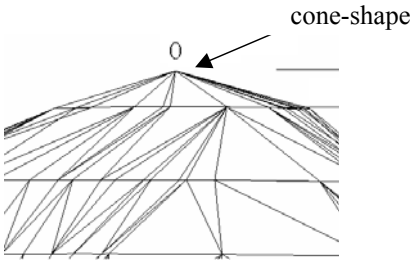


Fig. 8. Cone-shape closure of tissue

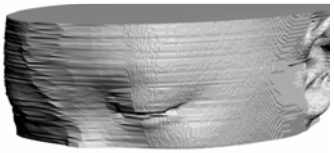


Fig. 9. Visualization of ear figure and surrounding

This study is mainly focused on the visualization of outer figure of tissues and organ. Based on the proposed methods, the topology structure of interior figures can also be obtained by changing the normal direction of vector of the resulted triangular plan, which can realize the visualization of the interior of tissue and organs.

5. REFERENCE

[1] J. Winder, R.S. Cooke, et al, “Medical Rapid prototyping and 3D CT in the manufacture of custom made cranial titanium plates”, Journal of Medical Engineering and Technology, 1999, Vol.23: 277-284.

[2] E.L. Hu, X.H. Xu, “Projection algorithm of fast light based on light relevant”, Journal of China Image Graph, 2004, Vol.9: 234-239.

[3] S.K. Laszlo, H. Vlastimil, et al, “On the efficiency of ray-shooting acceleration schemes”, Proceedings of ACM SCCG'02, Budmerice, Slovakia, 2002, 89-98.

A Face Recognition Algorithm Using Fast Fractal Face Image Compression

Yan Ma
Computer Science Department, Shanghai Normal University
Shanghai, 200234, China
Email: ma-yan@shnu.edu.cn
ShunBao Li
Computer Science Department, Shanghai Normal University
Shanghai, 200234, China
Email: lsb@shnu.edu.cn

ABSTRACT

A fast fractal face image compression algorithm based on symmetry and variance by adopting the inherent feature of facial symmetry is proposed. The best match domain blocks are constrained in the symmetrical region for a range block to speed up encoding. After analyzing the features of fractal codes, this paper defines the distance of fractal codes between images. Furthermore, a face recognition algorithm based on fractal codes is proposed. The experimental results on the ORL and YALE face databases show that the recognition rate of the proposed algorithm can achieve 88% and 92%, respectively. The results remain the same as that of the eigenface method and are higher than that of the variance based fractal-encoding method. The value of the coefficient in the algorithm is also analyzed.

Keywords: Fractal Encoding, Facial Symmetry, Fractal Codes, Face Recognition, Eigenface Method.

1. INTRODUCTION

In large face database, it is generally employed to compress face images firstly to save storage space and recognize face images directly in compressed domain in terms of speed. Recently, face recognition techniques in fractal domain are proposed. To achieve real time in recognition, it is first required to perform fast fractal compression. Jacquin [5] defined three types of blocks: edge blocks, midrange blocks and shade blocks, respectively. Lee [6] etc. proposed a variance based method. Truong [7] etc. proposed a fast fractal encoding method using space correlation. Based on the variance based fractal image encoding method in [6], a fast fractal face image compression algorithm is proposed using facial symmetrical feature. Furthermore, the features of fractal codes are analyzed and the distance of fractal codes between images is given. A face recognition algorithm based on fractal codes is proposed. The experimental results show that the recognition rate of the proposed algorithm remains the same as that of the eigenface method, and higher than that of the method in [6]. Compared with the eigenface method, the proposed algorithm holds its merit, that is, when a new query face image is imported into database, it is only required to perform fractal encoding on the new image without modifying other data in the database.

2. FAST FRACTAL FACE IMAGE COMPRESSION ALGORITHM

2.1 The variance based fractal image encoding

The original image I is partitioned into nonoverlapping range blocks R_i with size 4*4，8*8 or 16*16, etc. Then, I is partitioned into domain blocks D_i, which should be larger than that of the range block to fulfill the contractive requirement. To compare D_i with R_i, D_i should be contracted to the same size of R_i. Let $\{D_i\}(i=1,2,\cdots,n)$ be the domain pool. For each range block R_i, the encoder finds a domain block from $\{D_i\}(i=1,2,\cdots,n)$ to closely match it. The coefficients of the affine transformation together with the position of the domain block, which are the fractal codes of the image, are then stored for decoding, denoted it by $\{x_i, y_i, L_i, s_i, g_i\}$, where x_i and y_i are the coefficients to specify the position of the domain block D_i, L_i, s_i, g_i are the coefficient to specify one of the eight self-symmetrical transformations, the contractivity factor and the intensity shift, respectively. A fractal image-encoding algorithm based on variances was proposed in [6]. Calculate the variance $\{\sigma_{D_i}\}$ of $\{D_i\}$, and sort them in a descending order with their variances. For each R_i, calculate its variance σ_{R_i}. D_i is restricted in the region whose σ_{D_i} is $(w/2)\%$ higher and lower than σ_{R_i} with predefined w. This method can speed up encoding by reducing the number of domain blocks, yet it is for general image, not for specific face image. Based on the method in [6], a fractal image compression algorithm for face image is further proposed.

2.2 The proposed algorithm

In fact, fractal codes of a face image are a set of contractive mapping relationship from domain blocks to range blocks. The shape of domain block can be rectangle, square or triangle. The size of domain block can be constant or variable. The above variation will affect the final fractal codes. Even when the imported image has trivial variation, it will lead to the variation of partial coefficients of fractal codes. Thus, the region of domain blocks is constrained by using the feature of facial symmetry so that fractal codes can keep stability with the variation of face expression or pose.

Due to the symmetry of a face image (see Fig.1a), for a range block R_i, the best match domain block D_i can be found in its symmetrical region. Usually face image has certain variation in pose and expression; hence its symmetrical line does not lie in the center of face. While finding a best match domain block, the candidate region can be expanded to

region S_{i0}, see Fig.1c. If the best match domain block still cannot be found in S_{i0}, the candidate region can be expanded to region S_1, S_2 or S_3 respectively according to the position of range block R_i, see Fig.1d and 1e. That is, if range block R_i lies in the region S_1, the candidate region is expanded to region S_2. If range block R_i lies in the region S_2, the candidate region is expanded to region S_1. If range block R_i lies in the region S_3, which denotes that R_i lies in the center of face, the candidate region is expanded to region S_3.

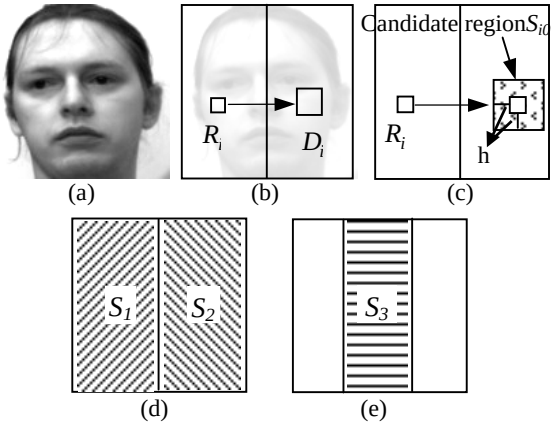


Fig. 1 The facial image and candidate region

Based on the above discussion, a face image compression algorithm is proposed as follows:

Assume the candidate region S_{i0} lies in the symmetrical position of range block R_i, and its size is h larger than that of R_i (see Fig.1c). Region S_1, region S_2 and region S_3 lie in the left side, right side and middle with $\frac{1}{3}$ width of the face image. Let threshold be T.

(1) Select one range block R_i from $\{R_i\}$ if $\{R_i\}$ is not empty and calculate its variance σ_{R_i};

(2) Denote the domain blocks by $\{D_{i0}\}$, in which σ_{D_i} of D_i is $(w/2)\%$ higher and lower than σ_{R_i};

(3) Extract D_i from $\{D_{i0}\}$ one by one and contract each domain block to the size of the range block R_i. Perform affine transformation on D_i and compare it with R_i. Once the distance between R_i and D_i is lower than threshold T, the domain block D_i is seen as the best match block and this mapped transformation coefficients are stored; Return (1);

(4) If the distance between R_i and D_i in the $\{D_{i0}\}$ is higher than threshold T, the candidate region is expanded to one of region S_1, S_2 and S_3 according to current position of R_i and it is satisfied that each σ_{D_i} in the candidate region is $(w/2)\%$ higher and lower than σ_{R_i}. Denote this candidate region by $\{D_{i1}\}$;

(5) Extract D_i from $\{D_{i1}\}$ one by one and contract each domain block to the size of the range block R_i. Perform affine transformation on D_i and compare it with R_i. Once the distance between R_i and D_i is lower than threshold T, the domain block D_i is seen as the best match block and this mapped transformation coefficients are stored; Return (1);

(6) If the distance between R_i and D_i in the $\{D_{i1}\}$ is higher than threshold T, the domain block D_i corresponding to the minimum distance in the $\{D_{i0}\}$ and $\{D_{i1}\}$ is seen as the best match block and this mapped transformation coefficients are stored; Return (1);

Under the worst case, that is, the best match block for every range blocks cannot be found in the candidate region S_{i0} and the candidate region is expanded to half size of the face image, the encoding time of the proposed algorithm is only half of that of [6], which explains that the encoding speed of the proposed algorithm is much higher than that of [6].

3. FACE RECOGNITION ALGORITHM BASED ON FRACTAL CODES

3.1 Fractal codes

In the fractal codes $\{x_i, y_i, L_i, s_i, g_i\}$, x_i、y_i represent the absolute position of a best match domain block for a range block, while it has no relationship with the position of the range block. If they are used directly, correct encoding and decoding can be got. While in the recognition, x_i、y_i should be replaced with d_i and θ_i [9] which is similar to polar coordinates, where d_i and θ_i denote relative distance and direction between domain block and range block, respectively. Small d_i indicates that the best match block is close to the range block, while large d_i indicates the best match block can be only found far from the range block. θ_i represents the direction of the best match block in terms of the meaning of angle. If most θ_i in the fractal codes are same, the image has the possibility of texture in certain direction. In addition, L_i reflects the relationship of eight self-symmetrical transformations among the best match blocks, while s_i, g_i reflects the difference in contrast for the image.

3.2 The distance of fractal codes

Let fractal codes corresponding to image I be denoted as follows:

$$IFS(I) = \{d_i, \theta_i, L_i, s_i, g_i; i = 1,2,3,\cdots,N\} \qquad (1)$$

The coefficients in Eq.(1) have already been normalized, where N is the total number of range blocks.

Let image I and J be of the same size, the distance between $IFS(I)$ of image I and $IFS(J)$ of image J is defined as follows:

$$\begin{aligned} &distance_{i,j} \\ &= \left\| IFS(I)_i - IFS(J)_j \right\| \\ &= \left\| d_i - d_j \right\| + \left\| \theta_i - \theta_j \right\| + \left\| L_i - L_j \right\| + \left\| s_i - s_j \right\| + \left\| g_i - g_j \right\| \\ &\quad \text{where}, \ i = 1,2,\cdots,N, j = 1,2,\cdots,N \end{aligned} \qquad (2)$$

The range of $\left\| d_i - d_j \right\|$ is between 0 and L, where L represents the diagonal length of image. The range of $\left\| \theta_i - \theta_j \right\|$ is between 0 and 2π. For $\left\| L_i - L_j \right\|$, the result is restricted to 0 or 0.5, that is, the result is 0 if L_i and L_j both are same, and the result is 0.5 if they are not same. The

range of $\|s_i - s_j\|$ and $\|g_i - g_j\|$ is between 0 and 1. Each term in Eq.(2) should be multiplied by different weight respectively in respect of different effect and range of the coefficients of fractal codes. So we get Eq.(3) as follows:

$$\begin{aligned} & distance_{i,j} \\ &= \|IFS(I)_i - IFS(J)_j\| \\ &= \|d_i - d_j\|/L + \|\theta_i - \theta_j\|/2\pi + \|L_i - L_j\| + \|s_i - s_j\| + \|g_i - g_j\| \\ & \quad where,\ i = 1,2,\cdots,N,\ j = 1,2,\cdots,N \end{aligned} \tag{3}$$

To speed up recognition, fractal codes $IFS(I)_i$ for a range block R_i in the image I is compared one by one with fractal codes for the region (see Fig.2) which is c larger than R_i, and the minimum distance is seen as the distance between $IFS(I)_i$ and image J, which can be denoted as follows:

$$\overline{distance_i} = \min\left(distance_{i,j}\right);\ j = 1,2,\cdots,N \tag{4}$$

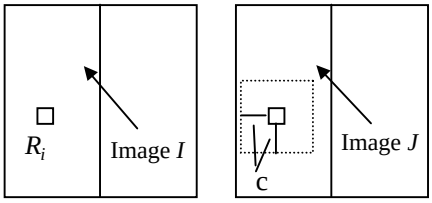


Fig. 2 The distance of fractal codes

According to Eq.(4), we can sum up the distance between each fractal codes of image I and image J. The distance of fractal codes between image I and image J is denoted as follows:

$$distance(I,J) = \sum_{i=1}^{N} \overline{distance_i} \tag{5}$$

3.3 A face recognition algorithm based on fractal codes

Suppose M face images and fractal codes $IFS(D_i), i = 1,2,\cdots,M$ for each face image are stored in face database D. Now given a query face image Q. A face recognition algorithm based on IFS fractal codes is proposed as follows:

(1) Calculate the fractal codes $IFS(Q)$ for the query image Q.

(2) Calculate $distance\left(Q, D_i\right), i = 1,2,\cdots,M$ according to Eq.(3)-(5).

The face image corresponding to the minimum distance from the results of (2) is seen as the recognition result.

4. EXPERIMENTAL RESULTS

All experiments are performed on 1.7 GHz PC with MATLAB. One is ORL face database, which includes 200 face images of 20 subjects, and each subject consists of 10 different images. We select 5 images (first 5 images) per subject from the database to construct a training image set (totally 100 images), denoted it by ORL-train. We select 5 images (the rest 5 images) per subject from the database to construct a recognizing image set (totally 100 images), denoted it by ORL-rec. Another is YALE face database, which includes 165 face images of 15 subjects, and each subject consists of 11 different images. We select 5 images (first 5 images) per subject from the database to construct a training image set (totally 75 images), denoted it by YALE-train. We select 5 images (the rest 5 images) per subject from the database to construct a recognizing image set (totally 75 images), denoted it by YALE-rec.

4.1 Recognition accuracy experiment

In the experiments, we employ the following methods respectively:

Method I: Train face images in the ORL-train and YALE-train by using the eigenface method. The number of eigenvectors is 40. Each face image in the ORL-rec and YALE-rec is imported to recognize. The face image corresponding to the minimum Euclidean distance is seen as the final result.

Method II: Employing fractal encoding by using the variance based method [6] for each face image in the ORL-train, ORL-rec, YALE-train and YALE-rec, where w is 30. Then, each face image in the ORL-rec and YALE-rec is recognized by using the proposed method, where c is 3.

Proposed method: Employing fractal encoding by using the proposed method for each face image in the ORL-train, ORL-rec, YALE-train and YALE-rec, where w is 30, h is 3 and threshold T is 0.1. Then, each face image in the ORL-rec and YALE-rec is recognized by using the proposed method, where c is 3.

Table 1 The recognition results of three methods

Recognition rate	Method I	Method II	Proposed method
ORL	87%	84%	88%
YALE	92%	89%	92%

From Table 1 it can be seen that the recognition rate of the proposed method remains same as that of eigenface method, which explains that the proposed method can achieve similar result as eigenface method. But one of the drawback of eigenface method is that the training time is too long for large face database, especially when a new query face image is imported into the training set, the eigenvectors have to be trained again. The proposed method can save storage space using the proposed fast fractal encoding technique, on the other hand, when a new query face image is imported into the training set, only this image is required to employ fractal coding. The recognition rate of the proposed method is obviously higher than that of method II, which is mainly due to the used fractal encoding method. Compared with the method of [6], the proposed symmetry based fractal encoding method restricts the best match domain block for a range block in its symmetrical region, which reduces the randomicity in the choice of domain blocks, so the performance in recognition rate and encoding time are improved.

4.2 The value of c

In the recognition process using the proposed method, the fractal codes of range block R_i in image I are required to compare with the fractal codes of the region c larger than R_i for image J, so the size of c affects the recognition rate and time. While c is too large, the recognition time will be increased. While c is too small, the speed of recognition will be improved and the recognition rate will be reduced. Table 2 shows the recognition rate for ORL and YALE database with different value of c. It can be seen that while c is larger than 3, recognition rate retains at 91% or 92%, the recognition rate will reduce to 80% or 84% with the decrease of c. So the value of c can set at 4 in terms of recognition rate and time.

Table 2 The recognition results with different c

The value of c	0	1	2	3	4	5	6	7
ORL (%)	80	82	84	88	91	91	91	91
YALE (%)	84	86	88	92	92	92	92	92

5. SUMMARY

Based on the variance-based fractal encoding method, this paper proposes the fast fractal face encoding method based on the symmetry and variance with consideration of the inherent symmetrical feature of face. Based on the above, the distance of fractal codes between images have been defined. Further, the face recognition algorithm based on fractal codes has been proposed. The experimental results on the ORL and YALE database show that the recognition rate of the proposed algorithm remains same as that of the eigenface method, higher than that of [6]. The suitable value of c is also given by the experiments.

6. REFERENCES

[1] Sharat Chandran, S. Kar, Retrieving faces by the PIFS fractal code, Sixth IEEE Workshop on Applications of Computer Vision, 2002, pp.8~12

[2] Tan T, Yan H, Face recognition using the weighted fractal neighbor distance, IEEE Transactions on Systems, Man, and Cybernetics-part c: Applications and Reviews, Vol.35, No.4, 2005, pp.582~590

[3] Ebrahimpour-Komleh H, Chandran V et al, Face recognition using fractal codes, Proceedings of International Conference on Image Processing, vol.3, 2001, pp.58~61

[4] Ebrahimpour-Komleh H, Chandran V et al, An application of fractal image-set coding in facial recognition, International Conference on Biometric Authentication, 2004, pp.15~17

[5] Jacquin A. E, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Transactions on Image Processing, 1992, Vol.1, No.1, pp.18~30

[6] C K Lee, W K Lee, Fast fractal image block coding based on local variance, IEEE Transactions on Image Processing, Vol.7, No.6, 1998, pp.888~891

[7] T K Truong, C M Kung et al, Fast fractal image compression using spatial correlation, Chaos, Solitons and Fractal, V22, 2004, pp.1071~1076

[8] S K Ghosh, Jayanta Mukherjee, et al, Fractal image compression: a randomized approach, Pattern Recognition Letters, 25, 2004, pp.1013~1024

[9] Y.Ma, S.B.Li, "Image retrieval based on IFS fractal code", Journal of Computer Applications，Vol.25, No.3, 2005, pp.594~595 (in Chinese)

[10] M Turk, A Pentland, "Eigenfaces for recognition", Journal of Cognitive Neuroscience, Vol.3, No.1, 1991, 71~86

[11] Y.Ma, S.B.Li, "Study on fractal image encoding based on the inner product of DCT coefficients and local variance", Mini-micro Systems，Vol.25, No.11, 2004, pp.2002~2005 (in Chinese)

A New Kind of Hybrid Model for Face Recognition

Weixin Chu
The School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, People's Republic of China
Email: chu_weixin@163.com
Hongwei Ge
The School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, People's Republic of China
Email: ghw8601@163.com

ABSTRACT:

Face Recognition has been widely studied and the literature dealing with face recognition is growing rapidly. In this paper, a new kind of hybrid model based on Support Vector Machine (SVM) and Hidden Markov Model (HMM) is proposed and has been used successfully in face recognition. The experimental results based on ORL face database show that compared with other methods, such as pure SVM and pure HMM, the hybrid model has the best performance.

Key words: SVM; HMM; Face Recognition.

1. INTRODUCTION

Face recognition has extensive applications such as identities validate, long-distance education, Human Computer Interface (HCI), and intelligent Robert Interface, etc. Recently, researchers have put forward many ways on face recognition. The traditional face recognition methods include eigenface, elastic template, neural network, and Hidden Markov Model (HMM). HMM is usually used for speech recognition, but it is first used for face recognition in reference [4], and now has been developed to embed HMM for face detection and face recognition [5,6,7,8]. However, in generally speaking, the rate of recognition based on these methods is not very satisfying, and those methods are not robust enough. Furthermore, they are sensitive to expression, illumination, angle, etc. So, we put forward a new way to solve this problem. HMM is good at dealing with sequential inputs, while SVM shows superior performance in classifying with good generalization properties especially for limited samples. Therefore, they can be combined to yield a better and effective method to recognize faces.

2. SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine is one of the latest and most successful statistical pattern classifiers that utilize a kernel technique as shown in reference [2]. The basic form of SVM classifier is summarized in the following.

The input vector $x \in R^n$ is expressed as

$$f(x) = \sum_{i=1}^{S} \alpha_i y_i K(u, x_i) + b \quad (1)$$

Where K is a nonlinear kernel function, S is the total amount of support vector. And x_i, y_i, a_i are i-th training sample, its class label, and its Lagrange multiplier, respectively, b is a bias.

SVM is founded on the VC dimension and the principle of SRM (Structural Risk Minimization). This principle considers: real risk is composed of two parts when we use a learning machine to classify the unknown data,

$$R \le R_{emp} + \sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

$$(0 < \eta < 1)$$

Where R is real risk, it is the risk boundary on the right of the inequation. R_{emp} is called experience risk. $\sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$ is called VC believable value. n is the number of training sample. h is VC dimension of the learning machine (it can reflect the complex degree of the learning machine). We can choose appropriate VC dimension to get a compromise between experience risk and believable value, and we will lower the real risk at last.

If they are nonlinear and inseparable data, we can use Cover theorem to mapping those data from low dimension space to high dimension space as in reference [3]. Kernel function can do this work efficiently. Generally speaking, there are three kinds of kernel function in SVM research, polynomial function, radial basis function and Sigmoid NN (Neural Network) function (we use sigmoid function in this paper).

3. HIDDEN MARKOV MODEL (HMM)

Hidden Markov Model is an effective model for sequential data such as speech signal. HMM was founded by Baum and Welch in 1960's.

HMM is composed of the elements which can be expressed as following:

(1) Total amount of the model state N. If S is the state set, then state of the model $S = \{s_1, s_2, \cdots\cdots, s_N\}$ is $q(t)$, and $q(t) \in S$, $1 \le t \le T$, where T is the length of the observation sequences.

(2) Initial state distribution Π, namely $\Pi = \{\pi_i\}$,

$$\pi_i = P[q_1 = s_i] \quad (1 \le i \le N)$$

(3) State transfer probability matrix A, namely

$$A = \{a_{ij}\}, a_{ij} = P[q_i = s_j | q_{i-1} = s_i] \quad (1 \le i, j \le N)$$

(4) State probability matrix B, namely $B = \{b_j(O_i)\}$, where O_i is observation vector when time is t. The

most commonly used probability function is:

$$b_i(O_i) = \sum_{k=1}^{M} c_{ik} N(O_i, \mu_{ik}, U_{ik}) \quad (1 \le i \le N) \ ,$$

where c_{ik} is the k-th mixed coefficient of the state i.

In brief, we can express a HMM as: $\lambda = (A, B, \Pi)$

4. FACE RECOGNITION BASED ON HYBRID MODEL

How to combine HMM and SVM effectively, take the most of advantages of the relative strengths of these two classification paradigms to improves the recognition performance of the system, is the main problem we should resolve. And the key problem is how to substitute MOG (Mixture of Gauss) with SVM in Hidden Markov Model. It is known to all that the output of SVM is distance, not probability. Fortunately, Platte has solved this problem in reference [1]. Firstly, he trained SVM C_i, and then, using Sigmoid kernel function, he got the probability of the unknown data x relative to C_i:

$$P(C_i \mid x) = \frac{1}{1 + \exp(Ax + B)} \qquad (2)$$

Where A and B are parameters of Sigmoid function. So, we can express the hybrid model as $\lambda = (Q, M, a, b, \pi)$, where Q equal 5, reflect the total amount of λ (five organs of a face); M equal 1, that is to say, to each state, there is a SVM corresponding to face organ basis character. a is a $N \times N$ matrix, denote the state transfer in the hybrid model λ, b is a $Q \times M$ matrix, denote the probability of observation of some kind of state, its value is $b = \{b_j(k)\}$.

$b_j(k) = P[o_t = v_k | q_t = j]$（$1 \le k \le M$）, denote when the state is j, the probability of O_t equal V_k. Its value can calculate by Eq. (2).

How to recognize faces with this hybrid model? The detailed approach can be expressed as following:

(1) Collect S faces which need recognized as training swatch. To each swatch, pick-up its basis organ character, train and create those organ's SVM.

(2) Use Baum-Welch algorithm, we get transfer probability parameter of HMM, and create the hybrid model S_i of a face, $S_i (1 \le i \le S)$, where S is the kind of faces need to be recognized. To each block B of faces, calculate the probability of B appear in the hybrid model S_i, namely $P(S_i \mid B)$.

According to formula Bayes, we have

$$P(S_i \mid B) = \frac{P(B \mid S_i) P(S_i)}{\sum_{1 \le i \le S} P(B \mid S_i) P(S_i)}$$

where $P(B \mid S_i)$ corresponds to the probability of B for a trained faces model S_i. Choose the S_i that makes $P(B \mid S_i)$ the greatest. Under this situation, we deem B is face S_i

5. EXPEREMENTAL RESULTS

The faces database ORL (the Olivetti and Oracle Research Lab's face image database) is used in our experiments. This database is composed of 40 people; each has 10 112×92 images. Each person has several expression and gestures as shown in figure 1.

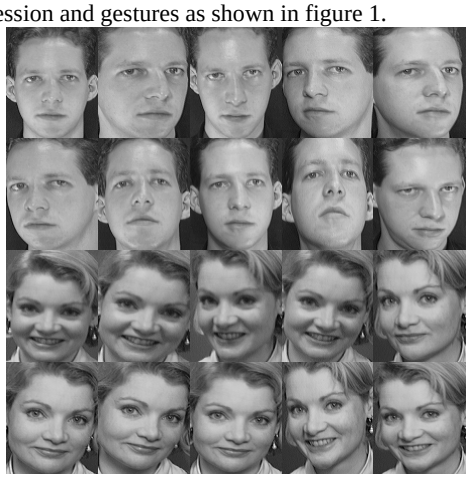


Fig. 1. Faces in ORL face database

We pick out 5 of 10 images as training sets, and the other 5 as test sets. So both training sets and test sets have 200 images. Each image is separated into five sections: forehead, eyes, nose, mouth, and chin. Pick-up basis vector character of every section, and then build SVM to recognize those organs. To full use topology restriction of every organ, we use a 5 states' Markov chain to restrain Recognition result. At last, we build a SVM/HMM hybrid model to recognize faces.

The experimental results are shown in table 1.

Table 1. the recognition results on ORL

Experimental method	Amount of swatch	Amount of Wrong	Rate of reject	Rate of recognition
Pure SVM	200	15	3.5%	92.5%
Pure HMM	200	18	4.5%	91.5%
SVM/HMM	200	5	2%	97.5%

The wrong recognition is because of some images (about 8%) of the database are not totally obverse faces, and result in we can't properly pick-up each organ basis character of these faces.

6. CONCLUSIONS

The method proposed in this paper is enlightened by Inline Markov Model[6]. But Inline Markov Model has some disadvantages, such as long training time and sensitive to expression, illumination, angle, etc. While using SVM/HMM hybrid model, the powerful discriminative ability of SVM and the ability of modeling the temporal property of sequential data are combined subtly. Detailed realization has been presented in this paper and experiments on ORL face database and Yale face database show that the hybrid SVM / HMM has better performance over the pure HMM and pure SVM .

7. REFERENCES:

[1] J.C. Platt, “Probabilistic Outputs for Support Vector Machines for Pattern Recognition”, U.Fayyad, Editor, Kluwer Academic Publishers: Boston, 1999.
[2] C.J. C. Burges, “A tutorial on support vector machines for pattern recognition”, Knowledge Discovery and Data Mining, 1998, 2(2).
[3] T. M. Cover, “Geometrical and statistical properties of systems and linear inequalities with applications in pattern recognition”, IEEE Trans. on Electronic Computers, 1965, 326-334.
[4] A.V.Nefian, M.H.Hayes, “Hidden Markov Models for Face Recognition”, in ICASSP99, 2721-2724.
[5] R. Chellappa, C.L. Wilson, etc, “Human and Machine Recognition of Faces: A Survey”, In Proc of the IEEE, 1995,87-111.
[6] B. Moghaddam, P. lexander, “Pentlandl Face Recognition using View-baced and Modular eigenspaces”, The International Society for Optical Engineering, 2003,1: 55-102.
[7] V. Blanz, T. Vetter, “Face Recognition Baced on Fitting a 3D Morphable Model”, Pattern Analysis And Machine Intelligence, IEEE, 2003,12:104-129.
[8] G.D. Guo, Z. Stan, “Face Recognition by Support Vector Machines Fourth IEEE International Conference on Automatic Face and Gesture Recognition”, 2000,6:213-257.

On Image Compression Using Evolutionary Computation And Swarm Intelligence

Yuping Chen
College of Information Technology, Southern Yangtze University
Wuxi, Jiangsu 214122, China
Email: cyp_wx@yahoo.com.cn

ABSTRACT

Advances in compression technique will have to keep pace with the exponential increase in the need for storage and transport of bulky multimedia images. In this paper，a method of based on Quantum-behaved Particle Swarm Optimization(QPSO)[7, 8] is proposed in order to realize the image compression problem and show a pack of graph with small bits, in view of dada is too vast and affect users to receive and use availably. Our experiment results indicate that the QPSO algorithm is a kind of efficient and reliable optimization arithmetic than Genetic Algorithms in resolving the image compression problem. The performance gain is especially significant for the image disposal.

Keywords: QPSO algorithms, Genetic algorithms, image compression

1. INTRODUCTION

The use and growth of multimedia technologies require that storage and retrieval of documents and images be expedited to meet response time constraints of end-users. Commercial full-length movies on a CD ROM could not exist without data compression. However, advances in compression techniques will have to keep pace with the exponential increase in the need for storage and transport of bulky multimedia images.

During the research of image compression in the past decades, such DPCM (Differential Pulse Code Modulation)、DCT(Discrete Cosine Transform)、D.A. Hoffman code、wavelet transform methods are advanced, and international compression standard based on proposed Discrete Cosine Transform like JPEG (Joint Photographic Experts Group) 、MPEG (Moving Pictures Expert Group) H.264 are lodged. However, some mistakes are appeared such as serious block effect in high compression rate 、eye vision effect characteristic which cannot be used in compression algorithms[2,3，4]. In this paper, a kind of Quantum-behaved Particle Swarm Optimization algorithms is introduced to resolve the ordered representation of image compression. Similar to genetic algorithms (GA), QPSO is a population based optimization tool. The system is initialized with a population of random solutions and searches for optima by updating generations. In QPSO, the potential solutions, called particles, are "flown" through the problem space by following the current optimum particles.

However, unlike GA, QPSO has no evolution operators such as crossover and mutation. Compared to GA, the advantages of QPSO are that QPSO is easy to implement and has few parameters to adjust. According to the results obtained in experiments we think that QPSO algorithms solution to the compression problem is very promising [9, 10].

* This work was supported by the National Natural Science Foundation of China under Grant No. 60474030

2. GENETIC ALGORITHM

The purpose of this paper is to provide a discussion of enhancing the compression rate of ensure that the QPSO has a good performance. The remaining part of the paper is structured as follows. In section 2, gives a brief of Genetic algorithms and concretely application in dealing with the ordered representation of compression. In section 3, gives explain of QPSO algorithms and particular illumination in graph compression problem in detail. And comparative experiment results and conclusions are showed in section 5、6.

GA is an exploratory procedure that can locate high performance structures in complex problem domains. It maintains a population of potential solutions, having a certain selection process based on fitness of individuals, and some recombination operators. A basic advantage of GA is the implicit parallelism, which makes these algorithms very fast and efficient. In this paper, the main problem is to analysis image compression using genetic algorithms based on the pixels of the image. We make use of genetic algorithms to obtain an ordered representation of the image and then perform the clustering to obtain the compression [2, 3].

2.1 GA algorithm is described as follows.

choose an initial population
determine the fitness of each individual
perform selection
repeat
    perform crossover
    perform mutation
    determine the fitness of each individual
    perform selection
    implemented clustering methods
until some stopping criterion applies

2.2 Fitness Function

We simply suppose to obtain an ordered element representation using the genetic algorithm, so what we intend to minimize is the disorder of them. In order to achieve this we propose: Given an ordered element representation: $X = [Xi]$， N ordered elements, where each element of the vector is another vector containing p bytes. The function is defined as follows:

$$F(xi) = \sum_{i=1}^{N-1} dist(x_i, x_{i+1}) \quad (1)$$

$$dist(x_i, x_{i+1}) = \sum_{j=1}^{p} (x_{i,j} - x_{i+1,j})^2 \quad (2)$$

Where:

The problem is now to minimize F, the distance sum from each element of the representation to the following one.

2.3 Parent Selection Operator

It is very important to select the parent particles, because the parent particles' characteristic determines the convergence velocity of whole evolution process. If the parent particles are super-excellence, the whole evolution process will get optimization in few generations, whereas，more time will be used. The problem we are facing is a function minimization, so the objective function f(x) must be mapped to a fitness function f(x). In this fitness function the probability of choosing a string xi to be crossed is given by:

$$\frac{f(x_i)}{\sum_{i=1}^{p} f(x_i)} \quad (3)$$

Where p is the population size. If f (x) is taken as the inverse of F (x) there will not be much difference between a good string and a bad one. This problem can be solved using another transformation

$$f(x) = C_{\max} - F(x) \quad (4)$$

Where C max corresponds to the value of the worst string in the population. Here, C max was computed like

$$\sum_{i=1}^{N-1} dist_{\max}(x_i, x_{i+1}) = (N-1) * dist_{\max}(x_i, x_{i+1}) \quad (5)$$

$$dist_{\max}(x_i, x_{i+1}) = \sum_{j=1}^{p} (\max(x_{i,j} - x_{i+1,j}))^2 \quad (6)$$

Each $x_{i,j}$ is a byte, then $\max(x_{i,j} - x_{i+1,j}) = 255$ (7)

and $C \max = (N-1) * \sum_{j=1}^{P} 255^2$ (8)

The problem encountered here was that due to the great value of the maximum constant computed for the fitness function, all the fitness valves for the chromosomes of a population were too close each other. This produces a generation evolution based practically on an equal probability basis and not based on the selection of the most suitable components of the old population. In order to solve the problem, the fitness function is scaled in following way:

$$f' = a * f + b \quad (9)$$

The constants a and b are determined using the following conditions:

$$a * f\min + b = 0, \quad a * fprom + b = fprom \quad (10)$$

solving this equations we obtained:

$$a = \frac{f_{prom}}{f_{prom} - f_{\min}}, \quad b = \frac{f_{prom} * f_{\min}}{f_{prom} - f_{\min}} \quad (11)$$

2.4 Cross Operators

In this crossing operation the first object in the offspring corresponds to the object in the first position of any of the parents. Then we determine the distance of the recently included object to the nearest one in both parents. The object that has the shortest distances is equal, and then one object is selected at random. This process continues until all the offspring positions are filled.

Let's see an example with 8 objects, in which the parents are: 1 = (5, 3, 8, 1, 2, 7, 6, 4) and P2 = (6, 5, 1, 2, 4, 7, 8, 3)

We take the first position in the offspring at random from the first position of any of the parents. Let's assume that in this case we select randomly P2, so the object 6 will be placed in the first position in the offspring. The objects nearest to 6 in parent P1 with a distance of 1 are {7, 4}. And in parent P2 is 5.

Therefore, we select at random on object from 4, 5 and 7. Let's assume that the one selected is 7, which will occupy the second position in the offspring. The closest objects to 7 in parent P1 and in parent P2 are 2, 6 and 4, 8 respectively. Because of the object 6 has already been included, we choose an object at random from 2, 4 and 8 for the third position in the offspring. Let's assume that the object selected is 2. In a similar manner all the remaining positions in the offspring are filled. The result chromosome is (6, 7, 2, 4, 8, 1, 5, 3).

2.5 Mutation Operator

This operator tries to solve some inherent problems of the genetic algorithms as regards local minima. Mutation process is that new offspring generate by cross-chromosome, which pass some genetic variation. The operator main ideal is that controlling object order of mutation position, and distance corresponding representation become small step by step. In fact, the genetic algorithms would be more useful in combinatorial optimization problems if they were adapted to invoke a local search strategy to optimize the final population members.

The mutation operator function selects at random two objects of the string and exchanges their positions.

2.6 Replacement Operator

This operator is used to select a fixed size population based on an old population P (old) and on the offspring P (offspring) which was created using the cross operators.

A parameter X, given by the user, determines that the new population P (new) has to be created using the X best strings from the combination of P (old) and P (offspring). The remainder of the strings from the strings from the new population P (new) is selected randomly from P (offspring). You can notice that when the parameter X is zero, the whole new generation is selected only from P (offspring). Otherwise, if the parameter X equals the population size then P (new) is formed by the best strings among all the strings in P (old) and P (offspring).

2.7 Implemented Clustering Methods

When the ordered element representation was obtained by means of the genetic algorithm, we proceed to make the clustering of it. With this purpose in mind, one method is explained below:

Fixed Clustering

This method is based on clustering the elements by assigning the same quantity of element per cluster. Once we have obtained the final ordered representation we divide the amount of elements by the amount of clusters, obtaining in this way the amount of elements that each cluster will contain.

$E = \frac{N}{M}$, where N = amount of elements, M = amount of clusters, E = amount of elements per cluster. The first E elements of the representation are assigned to the first cluster, the following E elements to the second one and so on until we assign the last E elements to the M cluster.

3. QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO)[1] is an evolutionary computation technique developed by Dr. Eberhart (http://www.engr.iupui.edu/~eberhart/) and Dr. Kennedy (http://users.erols.com/cathyk/jimk.html) in 1995, inspired by social behavior of bird flocking or fish schooling[5, 6]. However, the existing PSOs, make the particle only search in a finite space. So a Quantum-behaved Particle Swarm Optimization (QPSO) is proposed that outperforms traditional PSOs in search ability as well as guaranteeing global convergence algorithms. In the QPSO model[7，8，10], each individual is treated as a volume-less particle in the D-dimensional space. The particles move according to the following equations.

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1},\ldots\ldots,\frac{1}{M}\sum_{i=1}^{M} P_{id}\right) \quad (12)$$

$$p_{id} = \varphi * P_{id} + (1-\varphi) * P_{gd} \qquad \varphi = rand() \quad (13)$$

$$x_{id} = p_{id} \pm \alpha * \left| mbest_d - x_{id} \right| * \ln(\frac{1}{u}), u = rand() \quad (14)$$

Where, Vector $P_i = (P_{i1}, P_{i2}, \ldots\ldots, P_{id})$ is the best previous position (the position giving the best fitness value) of particle i called *pbest* , and vector $P_g = (P_{g1}, P_{g2}, \ldots\ldots, P_{gd})$ is the position of the best particle among all the particles in the population and called *gbest* . p_{id} , a stochastic point between p_{id} and p_{gd} , is the local attractor on the d*th* dimension of the i*th* particle, mbest is the mean best position among the particles. φ is a random number distributed uniformly [0,1]. u is another uniformly distributed random number on [0,1] and α is a parameter of QPSO that is called Contraction-Expansion Coefficient. In experiment the α is used by the equation:

α = (1.0-0.5)*(MAXTIMES-T)/MAXTIMES+0.5

3.1 Implementation of QPSO Methods

According to the pixels distributing of the image, every pixel is regarded as sample accordingly, by computing the pixels distance of Euclidean to perform the samples clustering. The distance equation of Euclidean is used according equation:

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \quad (15)$$

Here the problem is still to minimize distance $d(z_p, m_j)$. Unless the numbers of samples is small, it is difficult to search n samples' separation-degree. So we research a local-minimize position in the all-clustering distance by adjusting constantly the centroid m_j of the cluster and send the samples to the clusters whose centroids are nearest the samples according equation:

$$E = \sum_{j=1}^{c} \sum \left\| x_i - m_j \right\|^2 \quad (16)$$

Where E is the error of cluster (the summation of square-wind age)

3.2 The Quantum-behaved Particle Swarm Optimization (QPSO) algorithm is described as follows concretely.

for each particle
  Initialize an array of particles with random position
end
for T from 1 to MAXTIMES
clustering by the distance of Euclidean to equation
calculate fitness value to equation and determine the mean best position among the particles by

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1},\ldots\ldots,\frac{1}{M}\sum_{i=1}^{M} P_{id}\right)$$

for each particle
update the local optimization of particle *pbest* according equation (13), compare with the particle's previous best values: if the current value is less than the previous best value, then set the best value to the current value.
update the global particle *gbest* according equation (14), compare the current global position to the previous global: if the current global position is less than the previous global position, then set the global position to the current global
End
Repeat steps until a stop criterion is satisfied or a pre-specified number of iterations are completed.
end

4. Experiment Results

There are some parameters that affect the solution of GA and QPSO: population size (Psize), amount of chromosomes (Pchro), maximum iterations (Maxgen), and probability of mutation (Pmut), We have conducted extensive numerical evaluation of our implementation, across these parameter values, and have arrived at the following parameter values for the results reported in this paper: popsize = 30, pchro =3, pmut = 0.01, Maxgen = 100, small variations to these parameters values did not produce appreciable change in the quality of the final solution. In brief, we do not discuss some of these sensitivity aspects in this paper. Table 1. Below is organized to permit evaluation of GA and QPSO for a set of 10 typical color images. The images are obtained from a well-known public domain site*. Graph 1 is the fitness function comparison of GA and QPSO for ***plane*** image

representation.

As noted earlier, the quality of image (quantization error, see equation (15) of section 2) is used to compare the two algorithms. As can be seen from Table 1., the QPSO approach performs betters than GA approach for all data sets. The results of the testing are summarized in the table below:

Table 1. Comparison of the error in the iterative among particles between GAs and QPSOs

Image name	GAs	QPSOs	Change in %
blue hills	236.83	229.26	3.20
sunset	218.74	206.45	5.62
water lilies	230.48	214.36	6.70
winter	197.37	193.64	1.88
clb	224.47	203.14	9.51
misc	248.62	234.23	5.79
plane	214.47	192.30	10.34
lighthouse	249.74	233.85	6.36
face	236.91	225.51	4.81
caps	213.24	198.93	6.71

Above results indicate the comparison of complexity, *E* is the search-wind age. It is obvious that QPSOs search-distance below the GAs, accordingly curve is below.

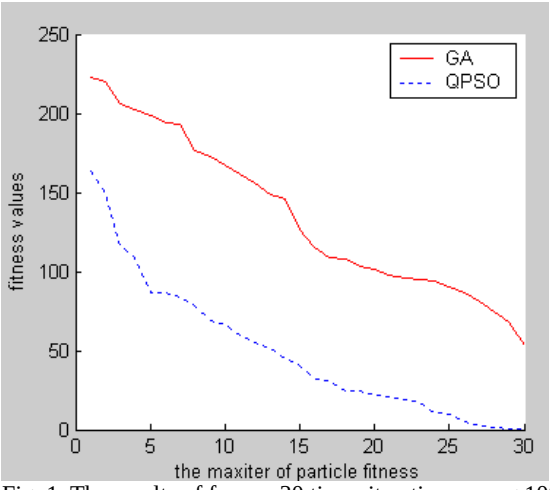


Fig. 1. The results of former 30 times iterative among 100 times iterative

5. CONSTRAINTS

The main constraint encountered in this work is that all the testing has been performed on the same image. In consequence , all the evaluations and performance comparisons among the different methods are valid in this context. We consider that it would be important to perform tests using other images with different sizes and characteristics to confirm or complete the results exposed in this paper.

6. CONCLUSIONS

The primary focus in this paper was on developing an algorithm whose performance quality is better than that of a well-known algorithm GA. Compare to GAs, QPSOs has no evolution operators such as crossover and mutation. In GAs, the pixels' optimization position is applied to the optimization representation.

However, In QPSO, the potential solutions, called particles, are "flown" through the problem space by following the current optimum particles.

The experimental results indicate that the Quantum-behaved Particle Swarm Optimization (QPSO) algorithm is simple in deal with the image compression problem, also reliable and effective to implement. In our future work, we will be devoted to find out a more efficient function to contract the compression, and therefore to improve the performance of QPSO fatherly.

7. FUTURE INVESTIGATION TRENDS

We think this work can be used as a basis to continue the research in image compression methods using QPSOs. In this section we propose some research trends that can be of interest:

·Implementation of the distributed computing methods by entering the compression percentage used as a parameter. In this way we would be the number of compression to be variable.

. Comparison of the performance of the different compression methods using different types of images, this research can complete or confirm the results of obtained in our work.

. Improvement of the QPSOs method, in which we can adjust a parameter control method based on population level and try to introduce a diversity guided, the method will intensify global search ability to win distinct effect.

8. REFERENCES

[1] J. Kennedy, R.C. Eberhart, "Particle swarm optimization", Proc. IEEE Intel conf. on neural networks Vol.3, IEEE service center, Piscataway, NJ, 1995, 1942-1948.

[2] Bhuyan, "Genetic Algorithm for Clustering with an Ordered Representation", Proceedings of Fourth International Conference on Genetic Algorithms, 1991,408-415.

[3] Klein, Dubes, "Experiments in projection and clustering by simulated annealing", Pattern Recognition, 1989, 22(1). 213-220.

[4] M. Ismail, Kamel, "Multidimensional data clusteringutilizing hybrid search strategies", Pattern recognition, 1989,22(1). 75-89.

[5] R. C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the sixth international symposium on micro machine and human science, IEEE service center, Piscataway, NJ, Nagoya, Japan, 1995, 39-43.

[6] R. C. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources", Proc. congress on evolutionary computation 2001 IEEE service center, Piscataway, NJ. Seoul, Korea. 2001.

[7] J.Sun, W.B.Xu, "A Global Search Strategy of Quantum-behaved Particle Swarm Optimization", Proceedings of IEEE conference on Cybernetics and Intelligent Systems, 2004,111-116.

[8] J. Sun, B. Feng et al, "Particle Swarm Optimization with Particles Having Quantum Behavior", Proceedings of 2004 Congress on Evolutionary Computation, 2004, 325-331.

[9] R. C. Eberhart, Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", Evolutionary-programming vii: proc. 7th ann. conf. on evolutionary conf., Springer-Verlag, Berlin, San Diego, CA., 1998.

[10] Y. Shi, R. C. Eberhart, "Parameter selection in particle swarm optimization", Evolutionary Programming, Springer-Verlag, New York, 1998. VII: Proc. EP 98 pp. 591-600.

Research of Error Resilience technology of compressed video based on IP Network

Xinnan Fan Xuewu Zhang Hua liu
College of Computer & Information Engineering, Hohai Univ.,
Changzhou, 213022, China
E-mail: fanxn@hhuc.edu.cn

Abstract

Video streaming over IP Network faces the challenges that the networks error-prone, transmission bandwidth is limits and fluctuating, the user device capabilities vary, unknown time delay, the networks are heterogeneous and so on. These challenges necessitate the need for smart adaptation the variable conditions of the network. This paper, we implement JSCC (joint source and Channel codec) based on minimum-distortion in order to deliver the best video quality given the network condition at the time of delivery. First, establish expectation distortion of decoder at the source terminal, at the same time calculate expect distortion of intra MB and inter MB. Second, establish two-state Markov chain for packet loss; calculate the probability of packet loss. Third, a global R-D optimization modes selection based on H.264/JVT is simulated by taking source distortion, channel distortion and the distortion of decoder into account. The simulation results show the error resilience strategies based on global R-D can enhancing QOS of video and enhanced ability of resisting on packet loss of IP network.

Key Words: H.264 standard, Error resilience mechanism, IP ,network jointly resource and channel coding.

1. Introduction

With the fast development of multimedia application, the request of its reliable communication is increasing. For compressed video information which transmit through network, because error will produced unavoidably during its delivery via network, which includes error codes, packet lost , it is needed to apply some error control methods to improve vision quality of the video information from decoder.

Joint source and channel error control mechanism is a perfect control mechanism. Its main idea is dynamic distribute bandwidth for coded video stream and FEC protection information of channel based on the global R-D theory, and improv the decoding quality of reconstruction video.

This paper proposed a new error control technique based on the new generation video compression standard H.264. It takes into account of behaviors of encoder, transmission channel and decoder synthetically and can achieve global optimization. This mechanism based on the R-D theory and combined the data partition package mechanism of H.264, can improve the ability of combating errors of coded video communication over IP network effectively.

2. Error Combating Mechanism Based on Global R-D Modes Selection

The typical R-D optimization mode doesn't consider packets lost of channel and the distortion of the receiver which imported by error concealment mechanism, so we proposed the global R-D optimization modes selection which consider source distortion, channel distortion and the distortion of decoder. If used Dr to denote R-D, Dr=Dq+Dep+Dc. In packets lost or error-prone environment, Dr is random variable. So the global R-D can be expressed by this formula[1，2]:

$$D = E[Dr] \quad (1)$$

Dq: the distortion produced by encoder quantization
Dep: the distortion produced by packets lost of channel
Dc: the distortion produced by error concealment of decoder

So the global R-D optimization modes selection can be defined by the follow formulas:

$$\min D(M_i^n) \quad , \quad R(M_i^n) \le Rc$$

where, $\min D(M_i^n)$ means the global R-D of micro-block F_i^n under the given mode M_i^n.

The global R-D can be expressed by SAD, MAD,SSD, MSE or PSNR. Here we select MAD

$$MAD = \frac{E\left\{\sum_{j=1}^{256} |\hat{f}ij - \hat{f}ij|\right\}}{256} \quad (2)$$

2.1 The Estimate of Decoder's Pixel Distortion by Encoder

There exits distortion because of the quantization and package strategy used by the encoder. Suppose used $F_{i,j}^n$ to denote original value of the jth pixel which in the ith MB of the nth frame and $\hat{F}_{i,j}^n$ denote reconstruct value in the encoder.

$$\hat{F}_{i,j}^n = \hat{e}_{i,j}^n + \hat{f}_{uv}^{n-1} \quad (3)$$

$\hat{e}_{i,j}^n$: Error of prediction

$\hat{f}_{uv}^{n-1}$: Can gained by motion compensation and

prediction of the previous frame

For the encoder, reconstruction value list $\hat{F}_{i,j}^{n}$ is random variable that should be measured by unbiased variance. Now, the expect distortion of the jth pixel which in the ith MB of the nth frame is:

$$d_{i,j}^{n} = E[(F_{i,j}^{n} - \hat{F}_{i,j}^{n})] = (F_{i,j}^{n})^2 - 2F_{i,j}^{n}E(\hat{F}_{i,j}^{n}) + E[(\hat{F}_{i,j}^{n})^2] \quad (4)$$

When calculating this formula, it is necessary to discriminate the pixel belongs to which MB intra frame or inter frame that should run recursive calculation respectively[3，4，5，6].

2.1.1 The estimate of expect distortion of intra MBs

If the packet which Intra($F_{i,j}^{n}$) belongs to received by the receiver correctly, then we can get this formula $\widetilde{F}_{i,j}^{n} = \hat{F}_{i,j}^{n}$ .The probability of this happening is (1-P), P denote the probability of data lost. If this packet lost, the decoder check the previous GOB whether it received correctly or not. If it received correctly, we use motion vector's mean value of near three MBs to establish relation between $\widetilde{F}_{n,v}^{n-1}$ and $\widetilde{F}_{i,j}^{n}$ We can get this formula $\hat{F}_{i,j}^{n} = \widetilde{F}_{n,v}^{n-1}$. The probability of this happening is P(1-P). If the previous GOB lost too, then the motion vector of MB which this GOB belongs to was set to 0, hence gets this formula $\widetilde{F}_{i,j}^{n} = \widetilde{F}_{i,j}^{n-1}$ .The probability of this happening is p^2.

Integrate these three cases above, we can get the expect value of $\widetilde{F}_{i,j}^{n}$ at the first and the second hour of $F_{i,j}^{n}$ which belongs to intra coding MBs[2，4，5，7.

$$\{\widetilde{F}_{i,j}^{n}\}_{intra} = (1-p)\hat{F}_{i,j}^{n} + p(1-p)E(\widetilde{F}_{i,j}^{n-1}) + p^2E(\widetilde{F}_{i,j}^{n-1}) \quad (5)$$

2.1.2 The estimate of expect distortion of inter MBs

Suppose the motion vector $F_{i,j}^{n}$ of MB which belongs to the nth frame can gain by predicting the previous frame $\widetilde{F}_{n,v}^{n-1}$, then the value of prediction can be expressed by $\hat{F}_{i,j}^{n-1}$, error of prediction is $e_{i,j}^{n}$ and error of quantization is $\hat{e}_{i,j}^{n}$. The reconstruct value of pixel at the encoder is $\hat{F}_{i,j}^{n}$. Hence we can get this formula:

$$\hat{e}_{i,j}^{n} = \hat{F}_{i,j}^{n} - F_{u,v}^{n-1} \quad (6)$$

In fact, the data is transmitted via network include $\hat{e}_{i,j}^{n}$ and motion vector. The probability that data packets transmits to the receiver correctly is (1-P). With the $\hat{e}_{i,j}^{n}$ and motion vector that received by the decoder and the reconstruct value of the jth pixel of the ith MB in the (n-1)th frame, it is easy to get the reconstruct value of the jth pixel of the ith MB in the nth frame by this formula: $\widetilde{F}_{i,j}^{n} = \hat{e}_{i,j}^{n} + \widetilde{F}_{i,j}^{n-1}$. For the encoder, $\widetilde{F}_{i,j}^{n-1}$ is random vector. Hence the follow formula comes into existence:

$$\{\widetilde{F}_{i,j}^{n}\}_{inter} = (1-p)E[(\hat{e}_{i,j}^{n} - \widetilde{F}_{i,j}^{n-1})^2] + p(1-p)E[(F_{u,v}^{n})^2] + p^2E[(\widetilde{F}_{i,j}^{n-1})^2] \quad (7)$$

The calculation above is processed by the encoder; the value of P can be achieved by detecting and calculating based on QOS monitor which monitor the sequence number of packets of RTP protocol at the decoder, then return this value to the encoder by RTCP protocol[3,4,5,8,9,10].

2.2 The establishment of Channel Mode of IP Network

Internet is a trying-best net of packet – switched. The packet lost which produced by network jam, buffer overflow and time–variable latency and bandwidth will induce the quality of video drop. So the data whether be received correctly or not can be modeled reasonably well with 2-state Markov chain.

Receive state (0): it denotes that one packet is received correctly in time.

Loss state (1): it denotes that one packet is lost because of network jam or exceed restrict of time.

p, q: transfer probability between the two states.

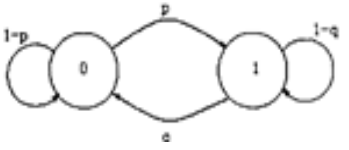


Fig.1 Two-state Markov chain for packet loss simulation

The transfer matrix of 2-state Markov chain can see the formula below:

$$A=\begin{bmatrix}1-p & p\\ q & 1-q\end{bmatrix} \quad (8)$$

$$P=\frac{n_1}{n_1-n_2},\quad q=\frac{n_3}{n_4}$$

n_1 is the number of successfully received packets when the previous packet is lost.

n_2 is the number of lost packets when the previous packet is lost.

n_3 is the number of lost packets when the previous packet is successfully received.

n_4 is the number of successfully received packets when the previous packet is successfully received[6，9，10].

3. Simulation Study of R-D Optimization Coding Algorithm of H.264/JVT Standard

There is a flexible and robust R-D optimization coding algorithm which is used in H.264/JVT standard. At the encoder, in order to improve the ability of error combating, first act simulation decoding respectively based on the K simulative states information of channel and gained expect distortion of the reconstruct frame of the decoder, then can achieve optimized selection of coding modes between MBs of video frames based on the R-D structure.

3.1 The Coding Mode-switch Based on R-D

The coding mode-switch of the R-D structure is run by confirming the coding mode of every MB can make the global distortion D be the least under a certain rate. So this problem can convert into the unrestricted Lagrange minimization:

$$J = D + \lambda R \quad (9)$$

λ is multiplier of Lagrange. The rate of code can be controlled by changing λ. For the global coding distortion, the effect of single MB can be accumulated. So we transfer this formula to solve the least distortion solely for every MB and the optimal coding mode of every MB can be choose by minimizing function.

$$\mathrm{MinJ}(F_i^n, M_i^n)=\mathrm{Min}(D(F_i^n, M_i^n))+\lambda R(F_i^n, M_i^n)) \quad (10)$$

We choose a simple one from the predecessor's methods of calculating λ to update the value of λ of every frame.

$$\lambda_{n+1}=\frac{2B_n+(\gamma-B_n)}{B_n+2(\gamma-B_n)}\lambda_n \quad (11)$$

B_n is the buffer which is occupied when coding the nth frame at the encoder. λ is the size of buffer when coding the nth frame. For every MB, it can achieve global R-D optimization by choosing coding modes and quantized steps reasonably[6,10,11,12].

3.2 The Configuration of Simulation Parameter

With analyzes above, we act simulation based on R-D optimization modes selection with the standard coder of H.264 (JM96). The key setting of configure file encoder.cfg see TABLE I and TABLE II. The version of the encoder is JM96, the test sequence is foreman.YUV and the format is QCIF, the frame rate of coding is 15FPS, the quantized steps of all the frames is 30 and the order of the coding is IPPP….., insert one I frame every 5 frames, number of referenced frame is 5, type of entropy coding is CABAC, a frame is partitioned into 9 slices, every slice have 11 16*16 MBs. After compression it can be packaged into 96 slices, every slice is packaged solely into the packet of RTP. The configuration of computer is PIV2.4G and 256M memory[6,13,14].

3.3 Simulation Result and Analysis

According to the setting of TABLE I and TABLE II, we run encoding and decoding simulation on 100 frame's foreman.YUV and analyze the influence that R-D optimization exerts on the coding ability under different channel.

TABLE I Coding parameters setting

ProfileIDC	88
LevelIDC	40
IntraPeriod	10
EnableOpenGOP	1
IDRIntraEnable	1
QPISlice	28
QPPSlice	28
FrameSkip	1
UseHadamard	
Motion SearchRange	16
NumberReferenceFrames	5
MbLineIntraUpdate	1
InterSearch16x16	1
Intra4x4DiagDisable	1

Table 2. Parameters setting of R-D optimization

RestrictSearchRange	2	
RDOptimization	2	
DisableThresholding	0	
DisableBSkipRDO	1	
SkipIntraInInterSlices	1	
UseExplicitLambdaParams	0	
LambdaWeightIslice	0.65	
LambdaWeightPslice	0.68	
LambdaWeightBslice	2.00	
LambdaWeightRefBslice	1.50	
LossRateA	10	Setting range (0-20)
LossRateB	0	Setting range (0-20)
LossRateC	0	Setting range (0-20)
NumberOfDecoders	30	Setting range (0-1000)
RestrictRefFrames	1	

Table 3. The experiment of R-D optimization under different channel

Lost rate of channel	P=0%			P=10%	
R-D optimization	RD=False	RD= True		RD= False	RD= True
		k=1	k=30	k=30	
PSNR of luminance of reconstruct video (dB)	35.91	36.44	36.44	30.44	33.92
Bit rate (Kbps)	154.01	368.48	368.48	154.01	369.507
Coding time (second)	109.594	246.375	509.717	109.221	507.00

From Table 3., we can see that under the error-prone channel, the R-D optimization made a raise about 1.4% in luminance's PSNR of the reconstruct video frame at the decoder, the bit rate of coding video sequence increased more than half and the coding time increased observably. At the same time, we can also see that with the different number of simulative channel state, for example, in the case of K=1 and K=30, luminance's PSNR of the reconstruct video frame at the decoder and bit rate of coding video sequence don't change but coding time changed observably. So the application of R-D optimization technique has improved the ability of encoder at the expense of coding time. So, when there are same states in K simulative channel states, only processing simulation decoding on the different states, this way, it will not effect coding ability and the result of R-D optimization and at the same time the coding time decreased observably.

In error-pro channel, lost rate is 10%, the R-D optimization technique improved the ability of combating error of coded video stream effectively and improve the vision quality of reconstruct video frame at the decoder. In this paper's experiment, luminance's PSNR of the reconstruct video frame has raised 10.24%. At the same time, it is clearly that the application of R-D optimization technique results in the increasing of bit rate of coded video stream and coding time. With analyzes above, we can make a summary that the application of R-D optimization technique improved the ability of coded video stream at the expense of higher bit rate and longer coding time.

4. Conclusions

This paper mainly studied the ability of error combating of video communication from three ways: source distortion, channel distortion and the distortion result of error concealment mechanism at the decoder and used R-D optimization technique to improve the ability of error combating of end-end video communication, and also analyzed some error combating techniques of H.264/JVT, then used test mode JM96 to test the global R-D optimization technique. The results show that end-end global R-D optimization technique can improve the ability of error combating of video communication, but this must at the expense of higher bit rate and longer coding time.

References:

[1] ISO/IEC 14496-2, "MPEG-4 video FGS v. 4.0,"N3317, Noordwijkerhout, NL, March 2000.

[2] G. S. Greenbaum, "Remarks on the H.26L project:streaming video requirements for next enerationvideo compression standards," ITU-T SG16/Q15,Q15-G-11, Monterey, CA, February 1999.

[3] B. Girod, N. F¨arber et al, "Scalable codec architectures for Internet Video-on-Demand," in Proceedings of 1997 Asilomar Conference on Signals andSystems, Pacific Grove, CA, USA, November 1997.

[4] J. Kimura, F. Tobagi, et al, "Perceived quality and bandwidth characterization of layered MPEG-2 video encoding," in Proceedings of the SPIE International Symposium on Voice, Video
and Data Communications, Boston, Mass, September 1999.

[5] M. van der Schaar , H. Radha, "Motion-Compensation Fine-Granular-Scalability (MC-FGS) for wireless multimedia," in Proceedings of IEEE Symposium on Multimedia Signal Processing (Special Session on Mobile Multimedia Communications), October 2001.

[6] F. Wu, S. Li et al, "A framework for efficient progressive fine granularity scalable video coding,"IEEE Transactions on Circuits and Systems forVideo Technology, vol. 11, no. 3, pp. 332–344, March 2001.

[7] H.264/AVC reference software, available online at:http://bs.hhi.de/~suehring/tml/download/.

[8] N.S. Jayant, P. Noll, Digital Coding of Waveforms,Prentice-Hall, Englewood Cliffs, NJ, 1984.

[9] K. Chang, R. Donaldson, "Analysis, optimization,and

sensitivity study of differential PCM systems operating on noisy communication channels," IEEETransactions on Communications, vol. COM-20, no.3, pp. 338–350, June 1972.
[10] N. F¨arber, Feedback-Based Error Control for RobustVideo Transmission, Ph.D. thesis, University Erlangen-Nuremberg, Germany, 2000.
[11] V. Varsa, M. Karczewicz, et al, R. Sj¨oberg,G. Liebl, and T. Stockhammer, "Common test conditionsfor RTP/IP pver 3GPP/3GPP2 - software and amendments," ITU-T SG16/Q6, VCEG-N37, Santa Barbara, CA, September 2001.
[12] T. Wiegand, et al., "Overview of the H.264/AVCvideo coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, 2003.
[13] G. Sullivan, T. Wiegand, "Rate-distortion optimization for video compression," IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74-90, Nov. 1998.
[14] T. Wiegand, "Rate-constrained coder control and comparison of video coding standards," IEEETrans. Circuits Syst. Video Technol., vol. 13, no. 7,pp. 688-703, July 2003.

Adaptive Multi-scale AMSS Operator for Quality Detection of Silks

Hui Cai, Jinhui Cai, Guangxin Zhang, Zekui Zhou
Department of Control Science and Engineering, Zhejiang University
Hangzhou, Zhejiang 310027, China
Email: caihui1982@hotmail.com

ABSTRACT

In order to better the silk image with multi-noise, a new improved filter operator based on affine morphological scale space (AMSS) equation was presented. The innovation of the new operator was introducing adaptive idea and scale space into partial differential equations. Then, aimed at testing the performance of the improved AMSS operator, a new evaluation method that could scale the effect of keeping edge information and ability of smoothing noise was presented. The experimental result illustrates that this improved operator can efficiently keep edge information, and the de-noising performance is more effective than other methods for the image with multi-noises.

Keywords: Image Processing, AMSS, PDEs, Noise Reduction, Quality Detection.

1. INSTRACTION

Silks have been one of the most important exports of Hangzhou. But recently, the texture of silks was influenced largely for the laggard technics and aging machines. Aimed at effectively detecting the flaws of the silk (such as added dots, irregular patterns, broken thread) and increasing benefits, computer vision detection system was introduced.
The presence of noises in images was inevitable. This could be introduced by the image formation process, or image recording or even image transmission process, which degraded the quality of the silk images and affected the exact identification of defects. Accordingly, how to filter the noises effectively was a crucial problem in the whole process. Image enhancement through noise reduction (also called image cleaning) is a fundamental problem in image processing [1]. A good filter algorithm should be de-noising sufficiently as well as keeping edge information effectively, which traditional algorithms usually couldn't fulfill them synchronously. Recently, an image processing method based on partial differential equation (PDE) became more and more popular for the performance of preserving details as eliminating noises. Now, PDE has been one of the most important tools in image analysis [2].

In the process of the detecting flaws on silk with computer vision, the silk images were always contaminated by multiple unknown noises. To minimize the conflict between reducing noises and preserving detail information, the notion of adaptive and scale space was combined with partial differential equation, and the affine morphological scale space operator (AMSS operator) based on adaptive multi-scale idea was presented. Then, a new evaluation method based on MSE and MAE was presented for quantitative analysis.

2. MULTI-SCALE AMSS OPERATOR AND ITS REALIZATION

2.1. The Concept of AMSS Operator

As a way of image preprocess, image filtering should satisfy contrast and affine invariance. The filter having these characters was corresponding to a battery of partial differential equations, and the more the limitations of invariance, the fewer the partial differential equations satisfied. While, the AMSS equation just had the characters of contrast and affine invariance [3], and could be mathematically given as:

$$\frac{\partial u}{\partial t} = |Du| \left(div \left(\frac{Du}{|Du|} \right) \right)^{1/3} \quad (1)$$

Where u was image; div was divergence operator; Du was the derivative of u. When t was very small, Eq.(1) could be re-written as:

$$\frac{u_t - u_0}{t} = |Du| \left(div \left(\frac{Du}{|Du|} \right) \right)^{1/3} \quad (2)$$

Where u_0 was the original image; u_t was the transformed image. This image transformation was called as AMSS operator [4].

2.2. Multi-scale AMSS Operator Based Iterative

The function of scale space operator was equal to solving differential equations. Accordingly, AMSS operator could be modeled as follows:
Assume that, $u_0(x, y)$ was a gray level image, if $(T_t u_0)(x, y) = u(t, x, y)$, then $u(t, x, y)$ satisfied the differential equation below:

$$\begin{cases} u_t = u_0 + t|Du| \left(div \left(\frac{Du}{|Du|} \right) \right)^{1/3} \\ u = u_0(x, y) \end{cases} \quad (3)$$

In this model, as the value of t was increasing, the number of the wave crests in the inflexion and curvature of image edge curve was decreasing monotonously. In this way, the extra features induced by disturbance were reduced largely, thus a high reliability was gained in the process of feature extraction. In addition, the scale parameter t could quantificationally reflect the degree of smoothness on images. In order to reach the smooth precision we need, the selection of parameter t was very important. But in the process of discretion the value of parameter t was restrained, because too big value would lead to biggish error and too small value couldn't satisfy the smooth precision. Therefore, a new filter algorithm based on small-scale steps and iterative was presented. Then Eq. (3) can be rewritten as:

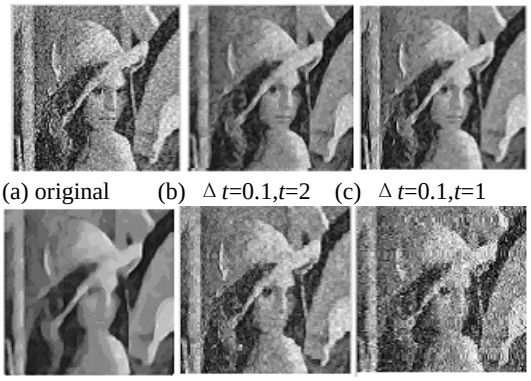
$$\begin{cases} u_{\Delta t} = u_0 + \Delta t \left| Du_0 \right| \left(div\left(\dfrac{Du_0}{\left| Du_0 \right|} \right) \right)^{1/3} \\ u_{\Delta 2t} = u_{\Delta t} + \Delta t \left| Du_{\Delta t} \right| \left(div\left(\dfrac{Du_{\Delta t}}{\left| Du_{\Delta t} \right|} \right) \right)^{1/3} \\ \vdots \qquad\qquad \vdots \\ u_{N\Delta t} = u_{(N-1)\Delta t} + \Delta t \left| Du_{(N-1)\Delta t} \right| \left(div\left(\dfrac{Du_{(N-1)\Delta t}}{\left| Du_{(N-1)\Delta t} \right|} \right) \right)^{1/3} \end{cases} \quad (4)$$

where Δt was scale step; N was iterative numbers; then scale parameter $t=N\times\Delta t$.

2.3. The Influence of Scale Parameter t upon Image Filter

In computer program, not analytical solution but numerical value was used to filter image. Therefore, a method based on finite differential was used for discretization of AMSS operator. In this way, it's easy and fast for solution. The details could find in reference.

The experiment was for standard image of "lena". The original image was contaminated by unknown noise. Assume that, Δt was scale step; t was scale parameter. The difference among iterative filters with different scale was illustrated in Fig. 1..



(a) original (b) Δt=0.1,t=2 (c) Δt=0.1,t=1

(d) Δt=0.05,t=2 (e) t=0.5 (f) without iterative t=1

Fig. 1. Iterative filter with different scale

In Fig. 1., (a) the original image; (b) image filtered by Δt=0.1,t=2; (c) image filtered by Δt=0.1,t=1; (d) image filtered by Δt=0.05,t=2; (e) image filtered by t=0.5 without iterative; (f) image filtered by t=1 without iterative. Obviously, the performance of small scale and iterative was better than filter without iterative. And it is also shown that, if t was invariable, then as Δt became lager, the smooth ability became stronger, but the edge information loss became severer simultaneously. While if Δt was invariable, then as t became lager the smooth ability became stronger too. But in the condition of high demanding for edge information, it was difficult to find the appropriate t for the image with multi-noises. Accordingly, adaptive idea was introduced to multi-scale operator.

3. ADAPTIVE MULTI-SCALE AMSS OPERATOR

It was concluded from the experiment above, as the scale or scale step became larger, the ability to reduce noise became stronger, and the loss of detail information became more. In this paper, firstly, Laplacian was introduced to discriminate noises from edges. Then, big scale and scale step were used for noise points, and small scale and scale steps were used for edge points. In this way, we could reduce noises efficiently and protect the edge information very well.

3.1. Noise Detection

Assume that, $\alpha(n1,n2)$ was the value of each pixel; $kp(p=1,2,3,4)$ were four one dimension Laplacians in different directions; kp was the template of 3×3, and the values were:

$$k1=\begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad k2=\begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

$$k3=\begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad k4=\begin{bmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Assume, $Jp=|\alpha(n1,n2)\oplus kp|$, $(p=1,2,3,4)$

Since in theory, the points of noise were considered as sharp changes in local region. If there was great disparity of gray level value between $\alpha(n1,n2)$ and all other adjacent pixels, Jp in four directions were all biggish, then $\alpha(n1,n2)$ was thought of noise. Contrarily, if $\alpha(n1,n2)$ wasn't noise, at least one of Jp $(p=1,2,3,4)$ was smaller.

3.2. The Model of Adaptive Multi-scale AMSS Operator

Assume that min $(n1, n2)$ was the minimal value of Jp, $(p=1,2,3,4)$, then the pixel coefficient defined as:

$$b(n1,n2)=\begin{cases} 1 & \min(n1,n2)^3 \geq T \\ 0 & \min(n1,n2)^3 \geq T \end{cases} \quad (5)$$

Where T was threshold. Then the filter based on adaptive multi-scale AMSS operator was expressed as:

$$u(n1,n2)=b(n1,n2)\times u1(n1,n2) + (1-b(n1,n2))\times u2(n1,n2) \quad (6)$$

Where u was the function of image filtered by adaptive multi-scale AMSS operator; $u1$ and $u2$ were separately the image functions in different scale spaces.

4. THE APPLICATION OF IMPROVED FILTER ON QUALITY DETECTION FOR SILKS

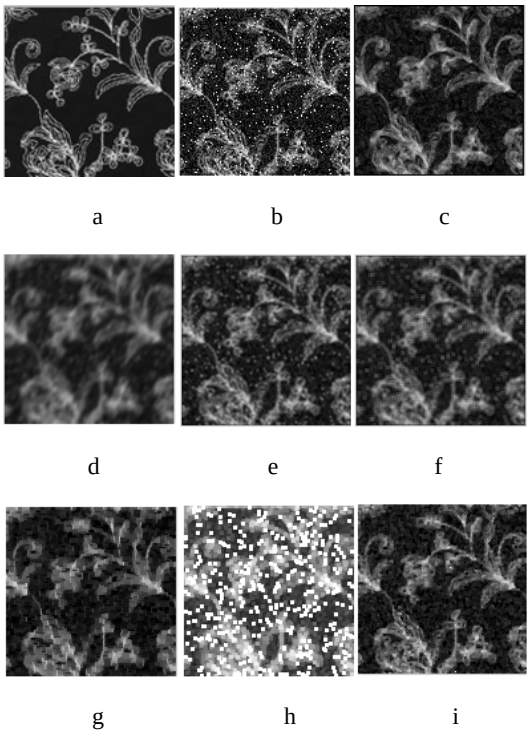
4.1. The Industrial Application of the Improved Filter

The requirement of detail information was very high in the quality detection for silk based on computer vision. But for the particularity of industry environment, lots of noises were induced into silk images in the process of image formation and transmission. And these multi-noises were very difficult to be modeling. Traditional methods always smoothed features when they were reducing noises [5]. Therefore, adaptive partial differential operator was introduced into image de-noising.

Fig. 2. was image processing for a piece of silk with fine

patterns. The parameters of the adaptive multi-scale AMSS operator were as follow: T=0.3，the parameters of the $u1$ was Δt=0.1，t=2，the parameters of $u2$ was Δt=0.05，t=0.5. In Fig. 2., (a) image required on ideal condition; (b) image with multi-noises; (c) image filtered by the improved AMSS operator; (d) image filtered by equation of heat conduction; (e) image filtered by weight average filter; (f) image filtered by average filter; (g) image filtered by open-close filter; (h) image filtered by morphological dilation; (i) image filtered by median filter.

According to the comparison of these methods, the filter based on equation of heat conduction (Fig. 2.,(d)) could smooth noises effectively, but it blurred the edge severely. So it wasn't able to meet the demand of silks flaw detection. Weight average and average filter (Fig. 2., (e,f)) were effective to Gauss noise, but the performance was bad for pepper and salt noise. The result of open-close and median filter was well, but consider both the ability to protect edge information and smooth noises, the method in this paper was better than them.



a b c

d e f

g h i

Fig. 2. Improved AMSS operator compare to other

4.2. Analysis of Variance for the Evaluation of Image Filter

There wasn't any uniform parameter for analyze the result of filter in image processing. Since the evaluation in 4.1 was based on visual effect, it was subjective. Accordingly, in this paper, two parameters that could value the edge loss and de-noising degree were solved separately for image evaluation. The two parameters were solved as follows:

1) Subtract the ideal image from the image with noises, then Get the image just including noises information through and marked with *A*.

2) Subtract the ideal image and A from the filtered image, then Get the image just including edge changes information in the process of filtering, and it was marked with *B*.

3) Get the MSE1 (mean square error) and MAE1 (minimal absolute error) of B separately.

4) Get the binarization of *A*, made it as template to filter the filtered image, then get the image including the noise information of the filtered image, and it was marked with *C*.

5) Get the MSE2 (mean square error) and MAE2 (minimal absolute error) of *C* separately.

Table 1. Evaluation

tag	MSE1	MAE1	MSE2	MAE2
c	0.0007	0.0072	0.0017	0.0053
d	0.0040	0.0323	0.0022	0.0060
e	0.0019	0.0191	0.0044	0.0099
f	0.0031	0.0263	0.0026	0.0073
g	0.1363	0.2619	0.0234	0.0234
h	0.0014	0.0102	0.0022	0.0060
i	0.0014	0.0127	0.0023	0.0063

The little the value of MSE1 and MAE1 were, the less the edge changed; while the little the value of MSE2 and MAE2 were, the less the noise in the filtered image had. Table 1. was corresponding to the Fig. 2..

It was shown that, for the image with multi-noises, the value of MSE1, MAE1, MSE2 and MAE2 were all the smallest in the image filtered by adaptive multi-scale AMSS operator. Therefore, in this way, edge changes and noises remained both were the least. The result was just consistent with the evaluation in 4.1.

5. CONCLUTIONS

Improving filtering technology, preserving details as well as removing noise had been pursued in image processing. Adaptive multi-scale AMSS operator was a new filtering operator based on PDE and combined adaptive idea and scale space. This operator didn't need the knowledge of noises, but utilized the structure information of the image [6]. Besides, in the algorithm, big scale and scale step were used for noise points, and small scale and scale steps were used for edge points. Therefore, the edge was preserved well in the image processing. Accordingly, it was equal to the silk quality detection in the process of which images were contaminated by unknown multi-noises.

6. REFERENCES

[1] A. P. Richard, "A New Algorithm for Image Noise Reduction Using Mathematical Morphology", IEEE Transactions on image processing, Vol.4, No.5, pp.554-568, 1995.

[2] S. J. Fu, Q. Q. Ruan, et al, "The application of PDEs in image processing", Computer Engineer and Application, Vol.2, 2005,pp.33~36.

[3] F. Meyer, M. Schmitt, Tribute to Georges Matheron, Springer-Verlag, 2005.

[4] T. Zhang, G. Chen, "Image Processing Based PDEs" High Educational Publishing Company,Beijing, 2004.

[5] L. S. Wang, Z. B. Xu. "The application of PDEs for the image analysis in biomedicine", Journal of mathematic engineering, Vol.21, No.4, 2004, pp. 475~490.

[6] Fedkiw, P. Ronald et al, "Shock capturing, level sets, and PDE based methods in computer vision and image processing: a review of Osher's contributions", Computational Physics, Vol.185, No.2, 2003, pp.309-341.

Wavelet Analysis and Processing Algorithm for Large-scale Image

Jianhua Zhang, Qianming Fu, Yan Ma
College of Information Technology and Software Engineering, Beijing Normal University Zhuhai Campus
Zhuhai, Guangdong, 519085 P. R. China
Email: zhuhaijianhua@yahoo.com.cn
And
Chunxia Lu
Guangdong Mobile Communication Co.LTD Zhuhai Branch
Email: lu-cx@21cn.com

ABSTRACT

Wavelet analysis, the newest time-frequency analysis tool with its unique advantages, was widely used in engineering. However, in the practical application, the huge size of the data made it difficult and impossible to process a Large-scale image by the ordinary wavelet process method. But Wavelet analysis still works well when the normal size image is processed. According to the facts mentioned before, we supposed that the batch treating data, distributional processing data, and the improvement wavelet algorithm might be the ways to solve Large-scale image's size problem. To prove our hypotheses' validity, we made a practical application contrasting the similar degree of the two large-scale images. Since we successfully get two images' error, we prove that one of our hypotheses-batch treating data is feasible and it is possible to use the wavelet analysis in the Large-scale image process.
Keywords: Wavelet Analysis, Large-scale Image Process.

1. INTRODUCTION

The wavelet analysis is recognized as the newest time-frequency analysis tool in the modern world. It is widely used in engineering like controlling system design, system emulate and so on. In signal and image process, other mathematical analysis methods (Fourier Transform or Short Time Fourier Transform) cannot substitute for the wavelet analysis in processing the non-steady signal, the discrete signal and other signals with time correlation.

2. WAVELET TRANSFORM

We analyze Fourier transform and wavelet transform separately.

2.1 Fourier Transform

Usually, image process is based on the Fourier Transform of 2-D, which definition is as the following.

$$F(u,v) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} f(m,n) e^{-jux} e^{-jvy} \qquad (1)$$

Where u and v are variables, $F(u, v)$ is called frequency domain of $f(x, y)$, the function of $F(u, v)$ are complex values, the period of u and v are 2π, anti-transform of formula (1) is as the following.

$$f(x,y) = \left(\int_{u=-\pi}^{\pi} \int_{v=-\pi}^{\pi} F(u,v) e^{-jux} e^{-jvx} du dv \right) / 2\pi \qquad (2)$$

We can't get any information about time in the transforming. In other words, because it cannot locate any position in time-domain, it suits not to analyze most of the signals that contain unstable amplitude-distortion or information of starting time and ending time.

2.2 Wavelet Transform

The definition of Wavelet Transform is as follow:

$$WT_x(a,\tau) = a^{1/2} \left(\int_{-\infty}^{\infty} x(t) \psi^*((t-\tau)/a) dt \right) \qquad (3)$$

Where $a>0$, Function $\psi(t)$ is called "the mother wavelet", after shifting τ, calculating in different scales, to decompose it into a series coefficients of wavelet functions as follows in frequency domain.

$$WT_x(a,\tau) = (a^{1/2}/2\pi) \left(\int_{-\infty}^{\infty} x(\omega) \psi^*(a\omega) e^{+j\omega\omega} d\omega \right) \qquad (4)$$

Where $x(\omega)$ and $\psi(\omega)$ are respectively the Fourier Transform of $x(t)$ and $\psi(t)$. Wavelet Transform can be interpreted as follows.

$$C(scale, time - position) = \int f(\tau) \psi(scale, \tau) d\tau \qquad (5)$$

If we multiply the coefficients gained above with related wavelet function scaled and shifted, and to superimpose their result separately, we can get the original signal. Wavelet Transform algorithm has its unique advantage in signal analysis. On one hand, we can analyze a signal with multi-resolution and multi-scale from various scale windows. On the other hand, after we select the appropriate basic wavelet function, WT has the capacity to express the character of partial signals in both time domain and frequency domain. However, Fourier Transform analysis and other methods haven't such features. Moreover, we can reduce processing data to enhance computing efficiency within certain scale with certain precision through Discrete Wavelet Transform according to Nyquist sampling theory. Additionally, Discrete Wavelet is based on the binary wavelet transform, that is, we should take the scale processing as 2^0, 2^1, 2^2, 2^3… and so on.

3. DIFFICULTY OF LARGE-SCALE IMAGE PROCESS

Ordinarily, the processing includes as follows.
Firstly, select a basis wavelet function and to decompose a sample image, and then compress its wavelet coefficients that should keep the characteristic of the original signal accurately. Secondly, decompose scale and select the best tree of decomposed construct through

algorithm analysis, then keeping the data of the best tree. Thirdly, we select a suitable basis Wavelet function. Since not every function can be used as the basis Wavelet function, only a few of applicable functions that can be used as the Wavelet function. On the other hand, aimed at specific analysis object, either comply with our experience or, test it many times, to select a suitable basis Wavelet function, we get the best effect. What is the best effect? The best effect is the outcome also decided by other condition, and is a fuzzy value that is balanced from various factors, even for the same object, different samples has different results.

After finished all the steps above, we start to handle the image decomposing process of the Wavelet Packet analysis that should be determined by the multi-resolution or the multi-scale. As is discussed above, generally speaking, the noise signals are almost mixed in the high-frequency details. Usually, we select a certain threshold to quantify noise in every scale in order to delete the noise signals; and this method is named noise elimination. Due to the differences of noise type, the threshold selection is still an ambiguity. So, selecting the threshold value is a complex process of fuzzy analysis and fuzzy calculation. After simplifying analysis, we can select a set value as threshold, and begin the noise elimination, which should from the most depth-position detail to previous scale, one by one and in turn.

With a common sense that the image compressing must maintain the characteristics of original image, we can decide a global threshold value and compressing the high-frequency section or select separately thresholds to compress in every scale. However, in order to get the approached anticipated result, the processing is slow and should be repeated many times.

All we mentioned above it is the image process method by the theoretically conventional wavelet analysis. In practical application, the processing large image is difficult or is simply impossible. As a meter of the fact that the large image containing the data quantity, which must be processed, is huge, the image contains more rulers and enlarges the sum of the each level high frequency coefficient. Also, in "two-dimensional separate wavelet analysis", it must use the addition of the large-scale matrix and the multiplication operates. Here, we have to say that both the huge size of participate operation data and the time the operating take cannot be imagined. Fortunately, here, it is unnecessary for you to worried anymore; since we offer you three methods we develop to solve such difficulties:

First: Before using the wavelet analysis image, make the division first to the large image so as to relatively reduce the area of the picture, which is going to process each time.

Second: We enhance the operation effect by the parallel computation, or carry distributional processing to the data. It makes the whole operation time reduced by using the network resources.

Third: We apply “the wavelet analysis combined with fuzzy inference for image process and arithmetic analysis”. To use the new algorithm makes the new wavelet function

4. THE PROPROCESS OF THE WAVELET ANALYSIS – IMAGE DIVISION APPLICATION

In the following part, we have a particular analysis to the pretreatment of the wavelet analysis -- Image division application and provide our experimental result. Our experimental goal is: through using the "2-D separate wavelet analysis" method, we make the pattern recognition and the analysis to two large images (Fig1 is the standard image and the Fig2 is sample image). Finally determine two images similar degree. Here, we separate our experiment into two mainly parts: The division part and the Wavelet analysis part.

Fig. 1. the standard image

Fig. 2. the sample image

4.1 The division part:

1) Step one: We use the wavelet function bior3.7 as the basis wavelet function in multi-scale decomposition image of "2-D separate wavelet analysis". Then make sure how large the image size is appropriate and ensure that the mass data matrix operation would be completed in the wavelet analysis in the certain time. This may be confirmed through the practice. We treat this image size as “measure scale” (actually it is limit to the greatest data quantity of the processed image) for preparing to measure other image sizes.

2) Step two: The result of measuring the standard image by measure scale decides how many parts the standard image will be divided. We suppose its value is N (make it to be integer for process advantageously). When the value of this data also is the circulation variable value, which we use to complete the circulation disposal procedure afterwards. The circulation variable of being

evaluated should not use the evaluation statement or read data from the data area; indeed, it should be evaluated through the keyboard entry—way during the program implementation. By this way, we may make the program to be more "general" and adaptable to different image processing size.

3) Step three: Divide the sample image into *N* part by our "division program". However, not every number can be divided into the multiplication form of two numbers, and sometimes it may be a prime number. In order to deal with the entire integer, we develop our original divide way. In our program, first, we find the biggest the number *M*, which not only close to *N* the most, but also, can be extracted. Second, we divide the number *N* in the addition form of two numbers; and ensure that one of them is the biggest number *P* which can be divided completely by *M*. And then we divide the picture into two rectangles according to *P* and *N-P*. By this way, even we sacrifice a column of the image; at least we meet the needs of the client. What's more, take the situation that the number *N* is huge enough for consideration; the significance of the last column is so little that it can be ignore completely. The only thing can't be ignore is the precision of the division. Since the type of the image we use in this program is 'TGA', which decides an image's size mainly by four variables, the coordinate of the start point, the length, and the width. It means that the more precise we get in these four variables, the more precise image we got. So, by decide the numbers of the digits after the radix point; we are able to assure our precision in the "division program". (In this application, the standard image is divided into four parts, as Fig. 3 shows us) And this is the first group of the image get from the "division program":

Fig. 3. 4 parts of the image

4.2 The Wavelet analysis part:

4) Step four: By the method of the bior3.7 basis wavelet function and "the 2-D discrete wavelet decomposition" we separately decompose the first group of image from multi-scale. And we get the four data from first group.

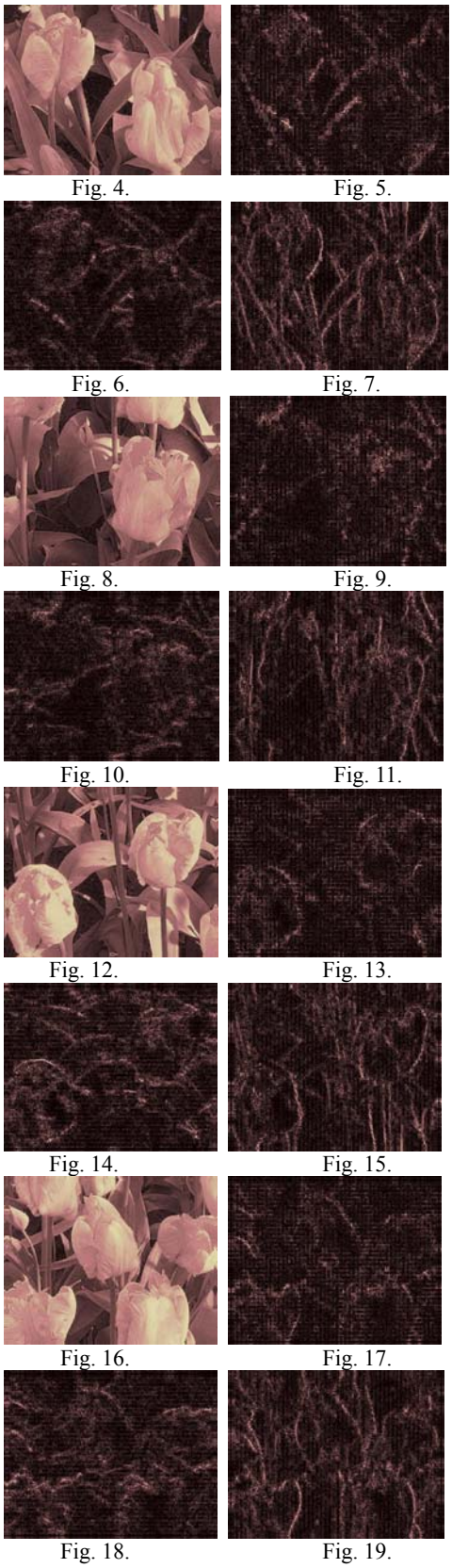


Fig. 4. Fig. 5. Fig. 6. Fig. 7. Fig. 8. Fig. 9. Fig. 10. Fig. 11. Fig. 12. Fig. 13. Fig. 14. Fig. 15. Fig. 16. Fig. 17. Fig. 18. Fig. 19.

Fig 4 to Fig 7 is recomposed images by the wavelet function. And Fig 4 is from the Approximation section. Fig 5 to Fig 7 is from the Detail section in the first layer. Fig 8 to Fig 11 is from the second part. Fig 12 to Fig 15 is from the third. And Fig 16 to Fig 19 is from the fourth.

5) Step five: Repeat step two to step four. We obtain the second group of image and the second set of the data from the standard image. As it must have a transformation during the practical application, people may concern about the loss. However, here is the reason why we choose the type of the image call "TGA" in our program, which contains over six kinds of image-file types and had been widely accepted by the international graphical images industries. What's more, since the percentage of the loss is fixed, if two of the images are almost the same, after the transformation, we still can get the same contrast result by compare the two results of the calculation.

6) Step six: Comparing each pair of the two sets of the data from the two images, and analyzing the first and the second data separately by the "energy function" of the wavelet analysis we obtained four groups of analysis results, which represent the similar degree by comparing the respective four parts of two-divided image. Table 1 is about the first data from the standard image. Table 2 is about the first data from the sample image. After the contrasting we get the errors from the two sets of the data. In the same way we can get the errors of the other parts.

Table 1. the First Data from the Standard Image

Name	Size	Bytes Class
A1	157x207	259992 double array
C	1x135681	1085448 double array
NbColor	1x1	8 double array
P	1x31	62 char array
S	4x2	64 double array
X	300x400	960000 double array
Xrgb	300x400	960000 double array
a	300x400x3	2880000 double array
ans	1x1	8 double array
cA1	157x207	259992 double array
cA2	86x111	76368 double array
cD1	157x207	259992 double array
cD2	86x111	76368 double array
cH1	157x207	259992 double array
cH2	86x111	76368 double array
cV1	157x207	259992 double array
cV2	86x111	76368 double array
map	255x3	6120 double array

The *cA1* is the Approximation section. The *cD1*, *cH1* and *cV1* are the Detail section in the first layer; *cA2* is the Approximation section and *cD2*, *cH2* and *cV2* are the Detail section in the second layer. By the same way we can get the data from the sample image.

Table 2. the First Data from the Sample Image

Name	Size	Bytes Class
A1	157x207	259992 double array
C	1x135681	1085448 double array
Ea	1x1	8 double array
Ed	1x2	16 double array
Eh	1x2	16 double array
Ev	1x2	16 double array
NbColor	1x1	8 double array
P	1x31	62 char array
S	4x2	64 double array
X	300x400	960000 double array
Xrgb	300x400	960000 double array
A	300x400x3	2880000 double array
Ans	1x1	8 double array
cA1	157x207	259992 double array
cA2	86x111	76368 double array
cD1	157x207	259992 double array
cD2	86x111	76368 double array
cH1	157x207	259992 double array
cH2	86x111	76368 double array
cV1	157x207	259992 double array
cV2	86x111	76368 double array
Map	255x3	6120 double array

7) Step seven: Analysis of the whole similar degree: The whole similar degree error of the two images is the average value of the four parts errors. Errors of the four parts respectively are 0.7028%, 0.7794%, 0.585%, and 0.7695%. So the whole similar degree of the two images is 0.7092%.

5 。CONCLUSION AND NEXT STEP WORK

CONCLUSION：

According the application instance above, we may say affirmatively that our method used in wavelet analysis of the large size image is completely feasible.

NEXT STEP WORK：

The above instance analyzes its similar degree by two images of the same size. When they are different size images, logically, we should apply a more complex method: Make the image zoom in or zoom out with proportion. And then compare the similar degrees of the two images. In order to achieve this goal, we need higher operating speed computers to reduce the whole operation time. Since Parallel algorithm and distributed computing has a significance place in image processing. The work that we are carrying on at present is "applying the distributed computing in the wavelet analysis to process large-scale data image" and improving the wavelet algorithm (compile new function) to enhance the operating speed.

REFERENCES

[1] *Application with wavelet Transform*, Science PRESS, Peking, 1999.
[2] Y.H. Pong, *Wavelet Transform and Engineering Application* (Science PRESS, Peking, 1999.
[3] Z.X. Cheng, Wavelet Analysis and Application, Xian Jiaotong University PRESS, Xian, 1998.
[4] http://www.mathworks.com.

Image Content Based Digital Watermarking Algorithm In Contourlet Transform Domain

LinGeng Guo1 ZhiBiao Shu2
College of Mathematics and Computer Science, Fuzhou University
Fuzhou, Fujian 350002, China
Email: 1superglg738@gmail.com ,2szb@fzu.edu.cn

ABSTRACT

A new extension to wavelet transform called Contourlet was introduced. It possessed not only spatial and frequency locality and multiresolution but also directionality and anisotropy. We made use of the trait of contourlet coefficient, and analyzed the embedding location and depth. The watermarks were inserted in locations, which were identified by contourlet coefficients corresponding to the strong image texture. And the embedding depth was adjusted intelligently based on the noise visibility function (NVF). The proposed algorithm is robust enough in withstanding some attacks such as filtering, adding noise, and JPEG compression.

Keywords: Wavelet, Contourlet, NVF, Digital watermark.

1. INTRODUCTION

As a valid method for secure identification and protection of electronic information, the digital watermarking technology obtains the widespread attention and the fast development. A reliable and secure digital watermarking system should well balance between the robustness and invisibility [1].

Due to favorable localization and multiresolution, the wavelets have been applied widely in the field of digital watermark. But, natural images are not simply stacks of 1-D piecewise smooth scan-lines; discontinuity points (i.e. edges) are typically located along smooth curves (i.e. contours) owing to smooth boundaries of physical objects. As a result of a separable extension form 1-D bases, wavelets in 2-D are good at isolating the discontinuities at edge points, but will not "see" the smoothness along the contours. In addition, separable wavelets can capture only limited directional information. In order to solve such problems, Minh N. Do proposed a double filter bank structure, named the pyramidal directional filter bank, by combining the Laplacian pyramid with a directional filter bank. The result is called the contourlet transform [2,3], which provides a flexible multiresolution, local and directional expansion for images.

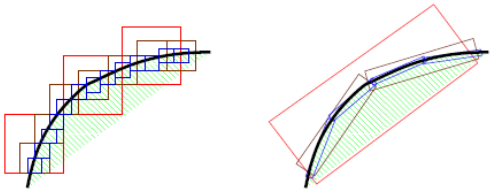
In this paper, we study preliminary application of contourlet transform in the domain of digital watermarking. And a digital watermarking algorithm based on image content in contourlet transform domain is presented.

2. CONTOURLET

2.1 Wavelet and Contourlet

Inspired by the painting scenario and studies related to the human visual system and natural image statistics, Minh N. Do identify a "wish" list for new image representations: multiresolution, localization, critical sampling, directionality, anisotropy [3].

Among these desiderata, the first three are successfully provided by separable wavelets, while the last two are solved by contourlet. As Figure 1 shows, because 2-D wavelets are constructed from tensor products of 1-D wavelets, the "wavelet"-style painter is limited to using square-shaped brush strokes along the contour, using different sizes corresponding to the multiresolution structure of wavelets. As the resolution becomes finer, we can clearly see the limitation of the waveletstyle painter who needs to use many fine "dots" to capture the contour. Contourlet style painter, on the other hand, exploits effectively the smoothness of the contour by making brush strokes with different elongated shapes and in a variety of directions following the contour [3]. Thus, contourlet can use much less coefficients to approach the curve than wavelets, and can provide the more sparse representation, especially to the images which consist of abundant edge and strong texture.



(a) Wavelet (b) Contourlet

Fig. 1. Wavelet versus Contourlet: illustrating the successive refinement by the two systems near a smooth contour, which is shown as a thick curve separating two smooth regions.

2.2 Contourlet Transform

Contourlet transform, viz. pyramidal directional filter bank (PDFB)[4], inherits anisotropy of curvelet transform. In a sense, it can be treated as another measure for curvelet transform, as shown in Figure 2. First, a multiscale decomposition into octave bands by the Laplacian pyramid (LP)[5] is computed, and then a directional filter bank (DFB)[6] is applied to each bandpass channel. Thus, the final result of contourlet transform is that using basis similar to contour segment to approach original image. Figure 3 shows examples of possible frequency decompositions by the contourlet transform. Figure 4 shows example of contourlet transform. It can be noticed that only contoulets that match with both location and direction of image contours produce significant coefficient.

3. APPLICATION OF CONTOURLET TRANSFORM IN DIGITAL WATERMARKING DOMAIN

* This work was partially supported by Fujian, China under Grant No.JB05047 and partially supported by Fuzhou University, China, under Grant No. 2005-XQ-15.

3.1 Watermarking Embedding Location Analysis

Although the algorithm that watermark is inserted in the low frequency coefficients is robust enough against attack such as compression, the human visual system is very sensitive to such modification. Contrarily, if watermark is embedded in the high frequency coefficients, watermarking algorithm is highly invisible but inferiorly robust. As shown in Figure 4, the middle frequency coefficients of contourlet transform characterize the local image properties, identifying textured and edge regions where the watermark should be more strongly embedded. So, it is feasible that the watermark is embedded in important locations of middle coefficients for a best tradeoff between the robustness and invisibility can be achieved.

Experiments show that the method can embed adaptively watermarks in strong-texture fields of original image, and achieve better effect than wavelet.

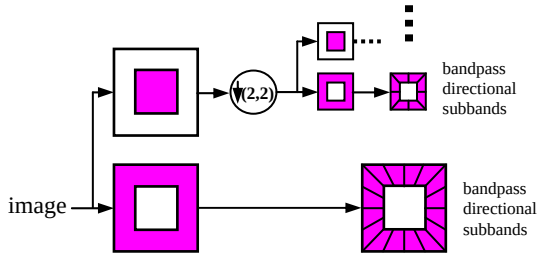


Fig. 2. The contourlet filter bank: first, a multiscale decomposition into octave bands by the Laplacian pyramid is computed, and then a directional filter bank is applied to each bandpass channel.

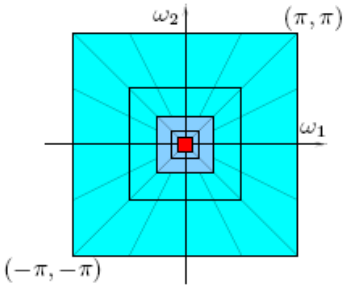


Fig. 3. Examples of possible frequency decompositions by the contourlet transform and

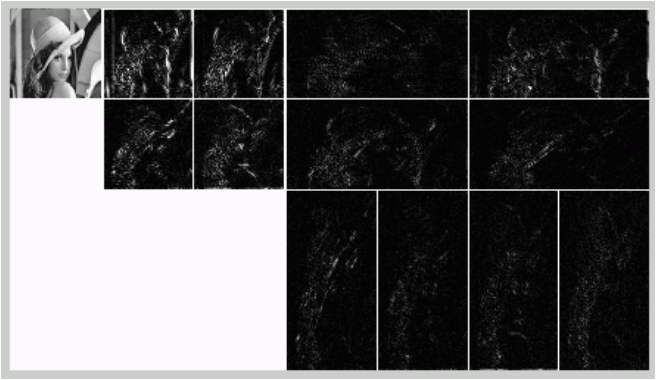


Fig. 4. Examples of the contourlet transform on the Lenna image. For clear visualization, this image is only decomposed into two pyramidal levels, which are then decomposed into four and eight directional subbands. Small coefficients are shown in black while large coefficients are shown in white.

3.2 Watermarking Embedding Depth Analysis

The watermarking embedding depth directly affects the visual quality and robustness of the watermarked image. The strong embedding depth leads to enough robustness against attack, but reduces visual quality. On the contrary, the weak embedding depth improves visual quality but reduces robustness. Thus, the best embedding depth should be automatically adjusted based on image content.

Let $CC_k = \{cc_{ij}, i \in [1, M], j \in [1, N]\}$, where CC_k denotes the coefficients of the k^{th} contourlet subband of size $M \times N$.

1) The embedding depth of the point (i,j)

Voloshynovskiy S proposed to relate the texture masking function to the noise visibility function (NVF)[7] as:

$$NVF_k(i,j) = \frac{1}{1+\theta_k \cdot \sigma_k^2(i,j)}$$

1)

where $\sigma_k^2(i,j)$ denotes the local variance of the image in a window centered on the point with coordinates (i,j); θ_k is a tuning parameter which must be chosen for every contourlet subband, and

$$\theta_k = \frac{D}{\sigma_{k\max}^2}$$

2)

where $\sigma_{k\max}^2$ is the maximum local variance for a given subband and $D \in [50,100]$ is an experimentally determined parameter.

2) The embedding depth of the k^{th} subband

$$\beta_k = 1 - std(CC_k)/a$$

3)

where **std** is an operation of computing standard variance, and a is an experimentally determined parameter.

3) The global embedding depth α

It is possible that watermarks are repeatedly embedded in more different subbands. Therefore, we define the global embedding depth α. It is an experimentally determined parameter too.

4. DIGITAL WATERMARKING ALGORITHM

4.1 Watermark Embedding

Figure 5 shows the flow chart of watermark embedding.

Let C' and C denote the watermarked image and original image respectively. The watermark is a random real series $P = \{p_1, p_2, \ldots p_n\}$ that match with $N(0,\sigma^2)$ Gaussian distribution. The watermark embedding is performed in contourlet coefficients as follows:

$$c'_{km} = c_{km} \cdot (1 + \alpha \cdot \beta_k \cdot (1 - NVF_{km}) \cdot p_m)$$

(4)

where c'_{km} is the watermark-protected coefficients, c_{km} is the n most former coefficients of the original middle-coefficient subbands, $m \in [1,n]$, k=1,2,3 corresponding to H12, H13, H14, and NVF_{km} is the corresponding NVF value.

At last, there are 3n coefficients, which are modified.

4.2 Watermark Detection

Let *C** and *WP* denote an attacked image with a certain type of attack and a certain watermark respectively. By the same contourlet transform, the subband coefficients c^*_{km} of the same embedding locations are achieved. Then the similarity between *C** and *WP* is defined in contourlet domain as follows:

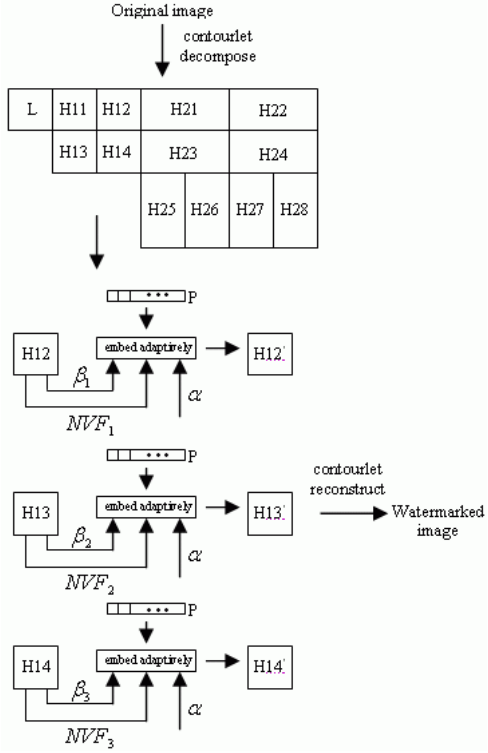


Fig. 5. the flow chart of watermark embedding

If original image exists, then

$$\rho_k(C^*, WP) = \frac{1}{n}\sum_{m=1}^{n} c^*_{km} \cdot wp_m \tag{5}$$

else

$$\rho_k(C^*, WP) = \frac{1}{n}\sum_{m=1}^{n} (c^*_{km} - c_{km}) * wp_m \tag{6}$$

.

According to [8], we define thresholds T_k as following:

$$T_k = 3.97\sqrt{\frac{2}{n^2}\sum_{m=1}^{n} c^{*\,2}_{km}} \tag{7}$$

The error which is produced by image distortion can be regarded as noise. In order to effectively restrain noise and improve precision of watermark detection, formula (5), (6), (7) can be improved as:

$$P = \prod_{k=1}^{3} \rho_k \tag{8}$$

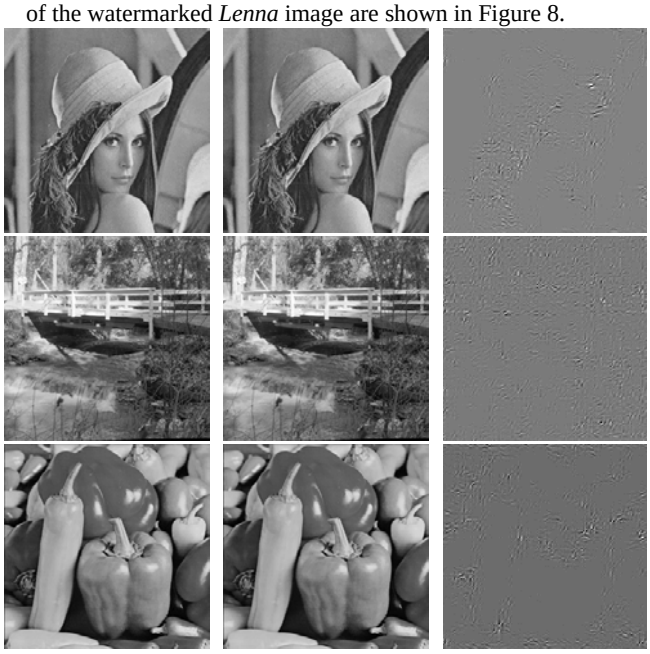
$$T_\rho = \prod_{k=1}^{3} T_k \tag{9}$$

Based on Neymann-Pearson rule, we say that the watermark is present if $P > T_\rho$, whereas $P < T_\rho$ if the decoder decides for the watermark absence.

5. EXPERIMENTAL RESULTS

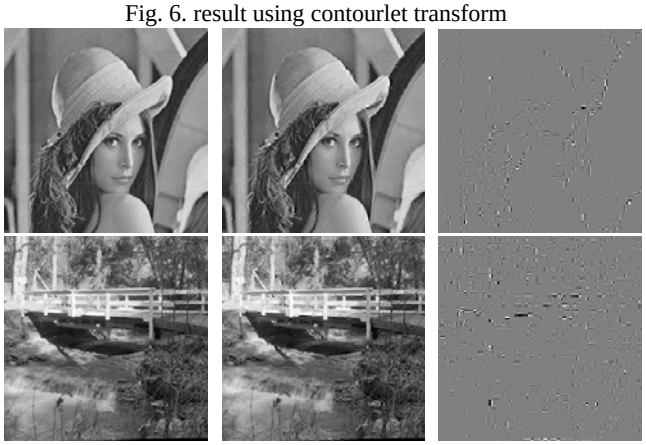
We make the separate experiments by contourlet and wavelet to test the algorithm of this paper. The watermarking series generated by the function of matlab6.5, and σ^2=10, *n*=1024, *D*=60. The experimental results are shown in Figure 6, 7, and Table 1. From top to bottom, the gray images of Figure 6, 7 are *Lenna*, *Bridge* and *Peppers* image of size 256×256. According to Figure 7 and Table 1, we can find that the watermarked image achieved from using wavelet transform method is of inferior visual quality and watermarks are embedded entirely in the curve edge of original image. As shown in Figure 6, however, because watermarks are inserted in the strong texture of original image, the visual quality of image achieved from the proposed algorithm is improved greatly.

For detecting the existence of watermarks, we randomly generate 500 different watermarks and the 250th watermark is the only true embedding watermark. The detecting results of the watermarked *Lenna* image are shown in Figure 8.



(a) original image (b) watermarked image (c) difference of (a) and (b)

Fig. 6. result using contourlet transform



(a) Original image (b) Watermarked image (c) difference of (a) and (b)

Fig. 7. result using wavelet transform

PSNR / mathod / name	contourlet	wavelet
Lenna	45.37	34.42
Bridge,	42.78	32.51
Peppers	45.43	33.73

Table 1. PSNR of contourlet versus wavelet

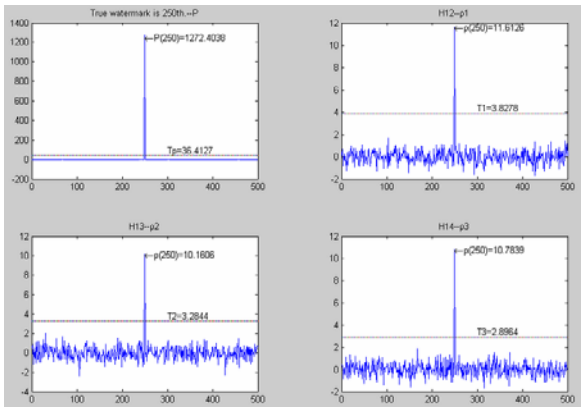


Fig. 8. Without attack, watermark detector response to 500 randomly generated watermarks. From left to right, top to bottom are: P_k, ρ_1, ρ_2, and ρ_3. Every peak value exists in the 250th location, and thresholds (T_p) are 36.41, 3.83, 3.28, and 2.90, respectively.

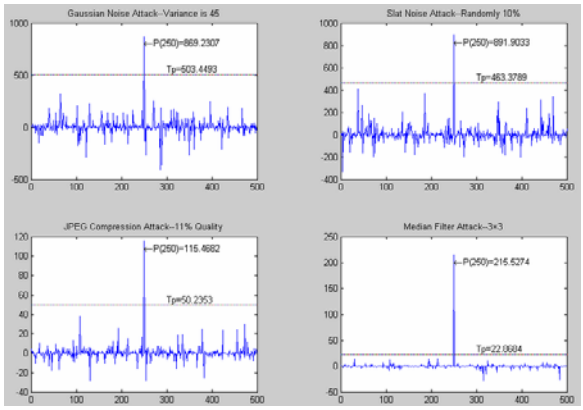


Fig. 9. Watermark detector response to 500 randomly generated watermarks. From left to right, top to bottom, the attacks are: additive Gaussian noise which variance is 45, additive 10% slat noise, JPEG compression with quality factor of 11%, and 3x3 median filtering. Every peak value exists in the 250th location and threshold (T_p) is 503.45, 463.38, 50.24, and 22.87, respectively.

Then, we generate some anamorphic images attacked by additive noise, median filtering, and JPEG compression. The detecting results according to formula (6) are shown in Figure 9. In those figures, the response to the correct watermark (i.e. No 250) is much larger than the response to the others and it is higher than the detecting threshold, thus showing the existence of watermark without doubt.

6. CONCLUSION

In this paper, we make use of the trait of contourlet coefficient, and analyze the embedding location and depth, then propose a digital watermarking algorithm based on image content in contourlet transform domain. The algorithm has some characteristics as:

1) The watermark embedding locations are adaptively identified based on the contourlet coefficients relating to the strong texture of original image.
2) Given the NVF, the watermarking embedding depth can be automatically adjusted based on the image content.
3) Analyze pertinence of detecting results, and improve conventional detecting method.
4) The original image is not necessary when detected.

At the same time, experiments show that the proposed algorithm using contourlet is better than using separable wavelets, and achieves a preferable tradeoff between the robustness and invisibility. Moreover, the proposed algorithm is robust enough in withstanding some attacks such as filtering, adding noise, JPEG compression.

7. REFERENCES

[1] C.H. Chang, Z. Ye, et al, “Fuzzy-ART based digital watermarking scheme”, 2002 Asia-Pacific Conference on Circuits and Systems (APCCAS’02), vol.1, 2002, 423-426.

[2] M. N. Do, M. Vetterli, “Contourlets: A Directional Multiresolution Image Representation”, Proc, IEEE Int. Conf on Image Proc., 2002, 357-360.

[3] M. N. Do, M. Vetterli, “The contourlet transform: An efficient directional multiresolution image representation”, IEEE Trans. on Image Processing, to appear.

[4] L.C. Jiao, S. Tan, “Development and Prospect of Image Multiscale Geometric Analysis”, Acta Electronica Sinca, 2003, 31(12A): 1975-1981.

[5] P.J. Burt, E.H. Adelson, “The Laplacian pyramid as a compact image code”, IEEE Trans. Commun, April 1983, 31(4): 532-540.

[6] R.H. Bamberger, M.J.T. Smith, “A filter bank for the directional decomposition of images: Theory and design”, IEEE Trans. Signal Proc., April 1992, 40(4): 882-893.

[7] S. Voloshynovskiy, A. Herrigel, et al, “A stochastic approach to content adaptive digital image watermarking”, Third International Workshop on Information Hiding, Dresden, Germany, September 29 - October 1st 1999, Vol. LNCS 1768 of Lecture Notes in Computer Science, 1999, 270-285.

[8] H. Li, Fowler, et al, “A Performance Analysis of Spread-Spectrum Watermarking Based on Redundant Transforms”, IEEE Int. Conf, vol.2, Aug. 2002, 553 -556.

A New Approach of the LED Lighting Scene Simulation

Peidong WANG
College of Computer Science & Technology, Harbin University of Science and Technology
Harbin 150080, China
Email: wpdlg@163.com
Qiubo ZHONG
College of Information, Northeast Forestry University
Harbin 150040, China
Email: zhongqiubo@yahoo.com
Xu Yang
College of Computer Science & Technology, Harbin Engineering University
Harbin 150001, China

ABSTRACT

There are many kinds of systems controlling the LED lamps, as the LED lamps are being used widely in the society. However, system simulating the scene of LED lighting is not supported. A new approach of simulation system including adding lamps, placement of lamps, adding snapshots for lighting, editing scene and downloading the scene data is designed for the LED lamps controlling system. In this simulation system, more than 256 kinds of color and five patterns of change of lamps are rendered. It is very easy and convenient for the user to design and simulate the scene of LED lighting by using this simulation system.

Keywords: OpenGL; Simulation; Texture Mapping; LED

1. INTRODUCTION

3D graphic technology is being used widely by simulation systems in field of the software and visible scientific calculation, compute simulation, virtual reality technology. However, few systems can simulate the scene composed of various snapshots of LED lighting. In the simulation systems of lighting, most can only simulate the intensity and illumination of lamps but can not simulate the scenes for the base control system to control the LED lamps more effectively.

According to the situation discussed above, we decide to design a new simulation system which can simulate the colors of lighting of lamps models and the patterns of changing of snapshots based on OpenGL embedded in VC++. In this new simulation system, the lighting will display more than 256 different kinds of color and there are many patterns of changing such as static, flicker and gradual change. The system will also have the function of editing the lamp model, editing the scenes and downloading the data of scenes.

2. BRIEF INTRODUCTION FOR OPENGL

OpenGL is defined as a software interface of the graphic hardware. In essence, it is a 3D graphic model library and has the strong transportability and fast speed. OpenGL is not like C or C++ which is a programming language. It is more likely a function library of running the C and provides some packing functions. In fact, there is not any program such as OpenGL. It is the code written by the programmer that uses the OpenGL as one of the API. Just like using the Windows API to access to the file, you can utilize OpenGL to create the graph.

3. SIMULATION ALGORITHM

This system simulates the process of adding the lamps models, placement and scene play. At first, we should define the scene model composed of different lighting, and then, set the selecting algorithm of lamps model and the snapshots based on the display of color of lamps model and the pattern of changing. Finally, the scene is played according to dividing frame algorithm by scene player.

3.1 Scene model

The object of this simulation is the LED lamp control system, so it primarily simulates the color of lighting and the pattern of changing.

We suppose $Pij(x)$=Q(F_C,B_C,F_T) to present the pattern of changing of number j snapshot of number i lamp, where F_C presents the front color, B_C presents the back color and F_T presents the flicker rate. The x has three values as follows:

1. when x=0, F_C=B_C; F_T=0 it presents the pattern of static;

2. when x=1, F_C≠B_C; F_T=0 it presents the pattern of gradual change;

3. when x=2, F_T=r and r is random it presents the pattern of flicker.

We also suppose function $Hi(j)$ to present the duration of number j snapshot of number i lamp. The formula as bellow:

$$\sum_{j=0}^{N} Hi(j) = Hi(1) + Hi(2) + \ldots + Hi(N) \quad (1)$$

It presents the sum of all duration of N snapshots of number i lamps, where the all duration of all snapshots of any lamp should be equal. $\sum_{j=0}^{N} Hi(j) = \sum_{j=0}^{N} Hp(j)$;

where i, p∈ (0,1,2,...)。

Then we suppose Sij to present number j snapshot of number i lamp and therefore can get the formula as bellow:

$$Sij = Pij(x) + Hi(j) \quad (2)$$

If we suppose Li to present number i lamp and

get the formula as follows: $Li = \sum_{j=1}^{n} Sij$ (3)

Formula (2) presents the sums of all the N snapshots of number i lamp. If we put the formula (2) into formula (3), we will get the formula (4) as bellow:

$$Li = \sum_{j=1}^{n} \left(Pij(x) + Hi(j) \right) = \sum_{j=1}^{n} Pij(x) + \sum_{j=1}^{n} Hi(j) \quad (4)$$

Finally, we suppose Mr to present number r scene, so we can get $Mr = \sum_{i-0}^{t} Li$ to present the sum of t lamps of number r.

The whole LED lighting scene is composed of many frames of scene. Each scene is composed of many snapshots and each snapshot is composed of the state of every LED lamps' change pattern. The pattern of change can be composed of static, gradual change and flicker. The whole scene model set the LED lamp as a unit and the duration of every snapshot of every LED lamp can be different. For every LED lamp, its state of snapshot can be composed of several stable frames just as the film we seen in the cinema.

3.2 Coordinate conversion

In the OpenGL, the coordinate provided for the location of OpenGL is called as Object Coordinate. The Object Coordinate uses the world coordinate and the general coordinate given by the screen is 2D coordinate. The world coordinate is a right hand coordinate system. In the OpenGL x axis is from left to right, y axis is from down to up and z axis is from in to out. Since the simulation system simulates a 2D space, we suppose the coordinate of z axis to be 0. For the zoom of the space, we do the zoom conversion according to the factor given. It spreads all the locations in the orientation of three axes to increase the scale of objects. The function glScalef(GLfloat x, GLfloat y, GLfloat z) multiply x, y and z using zoom factor given. The zoom of course does not need to be consistent. However, our lamp model is a circle so the zoom should be consistent. Because the simulation is in 2D space, the zoom in the orientation of z axis is a fixed constant.

In the process of moving the lamp model, we are only interesting in the coordinate of x and y axes because the simulation system is in 2D space. We do not consider the coordinate of z axis and so what we need only is to get the new coordinates of screen of the lamp model after it moved by the mouse.

3.3 Algorithm for selecting lamp model

We adopt the minimum distance method for selected the lamp model by mouse. First, it gets the coordinates of mouse on the screen (x, y), and then, judges whose distance between the lamp coordinates（GetPosX()，GetPosY()）and the coordinates of mouse on the screen is the shortest from the queue of the all lamp models. The distance between the coordinates of lamp and the coordinates of mouse on the screen can be calculated by the formula (5) as bellow:

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5)$$

The algorithm for selecting lamp model in C program is described as bellow:

```
for(all the lamp models in the queue)
{
dx=(pLight->GetPosX()-x)×(pLight->GetPosX()- x);
dy=(pLight->GetPosY()-y)×(pLight->GetPosY()- y);
D = (dx + dy) 2 ;
if(D< dMin)
{
      dMin= D;
      pTmp= pLight;
}
}
if(dMin <fixed constant)
{
data.index= pHelper->GetLightIndex(pTmp);
it is this lamp model that is selected by the mouse;
}
```

3.4 Color display and change of the lamp model

It requires certain of the simulation time and the effect of simulation because this simulation system simulates the scene composed of different lightings of lamps and mainly simulates the effect of moving scenes, which are composed of colors change and patterns of change of lamps. However, it does not require highly real of simulation of the lamps. According to these features of simulation discussed above, we control the simulation system by using texture mapping of OpenGL, and a conversional function converts the color of lighting of lamp from 32 bits for windows to 256 kinds consisting of RGB.

According to different patterns of change, gradual change including gradual change from darkness to brightness and gradual change from brightness to darkness and static can be controlled by the same method, and flicker should be controlled by another method. The format of snapshot contains four bites messages described as bellow:

F_C	B_C	D15 D14 D13 D12	D7 D6 D5 D4
		D11 D10 D9 D8	D3 D2 D1 D0

F_C represents front color of lighting of lamp and B_C represents back color of lighting of lamp. The fifteenth bit (D15) of the third bite is a mark bit which represents whether the pattern of change is flicker. According to the different value of D15, there are two situations bellow:

1. D15=0; gradual change, static; D14…….D0 represents the duration.
2. D15=1; flicker; D14……D9 represents the times of flicker; D8……D0 represents the duration of flicker.

The pattern of gradual change and static calculate the increment D of data required to be sent to the display class of the simulation system every time according to the following formula (6):

D=(BACK_COLOR-FRONT_COLOR)/(LASTING_TIME/20ms) (6)

Where FRONT_COLOR represents the value of front color, BACK_COLOR represents the value of back color and DURARION represents the duration of snapshot. The quantum of CPU is 20ms. So the time of CPU scanning in one snapshot is calculated by the formula described bellow:

TIMES = LASTING_TIME /20 (7)

The increment value for pattern of change of static is 0. After increment D is sent to the display class of the simulation system every 20ms by TIMES numbers, it can achieve to control the gradual change and static pattern of change of lamb models.

The pattern of flicker calculates the duration of snapshot by formula (8) describe bellow:

DURATION =FLICK_TIMES×FLICK_TIME (8)

Where FLICK_TIMES represents the times of flicker and FLICK_TIME represents the duration of flicker. The values of the FRONT_COLOR and BACK_COLOR are sent to display class of simulation system every the time of FLICK_TIME alternately, and after the data was sent for FLICK_TIMES times, it will finish controlling the pattern of change of flicker.

3.5 Algorithm for scene play

1. Initialize the environment.
2. Read the data of snapshot and judge the pattern of change.
3. If the pattern of change is flicker, then calculate the duration LASTING_TIME by the formula (8) and turn to step 4. If the pattern of change is other pattern, then calculate the increment D and the times of CPU scanning TIMES and turn to step 5.
4. Change the state of snapshot according to the duration and the flicker times FLICK_TIMES and duration of flicker FLICK_TIME presented in the data bite calculated and then turn to step 6.
5. Change the data of snapshot to add the increment D every time of CPU scanning and then turn to step 6.
6. Judge whether the snapshot is finished playing. If it does then turn to step 7, else turn to step 2.
7. Judge whether the scene is finished playing. If it does not, then turn to step 2.
8. End.

4. SYSTEM DESIGNNING BASED ON OPENGL EMBEDDED IN VISUAL C++

OpenGL programming is just like C programming and in fact the interface of OpenGL is C. Therefore, the programs in many books on OpenGL call the standard (or Win32 console Application) functions of OpenGL to get achievement. To do this let us know the basic operations of OpenGL quickly. However, OpenGL only contains more than 120 libraries of graphic functions but not contains the windows functions. Although the auxiliary library of OpenGL provides some windows and event management functions, these functions are not enough in the real applications. It lacks the ability of object oriented and which does not conform to the ideas of software design under the popular circumstance. So we require a help of “windows” system to achieve the graphic work of OpenGL. MFC in VC++ contains very strong application frames based on Windows and provides rich windows and events management functions. The MFC has become the popular work platform now. Therefore, we can use MFC to call OpenGL functions to achieve the graphic work.

4.1 Structure of simulation system

This LED lighting simulation system has a very kind man-machine interface. The user can operate all the functions of the simulation system only by mouse and keyboards. There are four subsystems consist of the whole system. The structure frame of system is displayed in Figure 1.

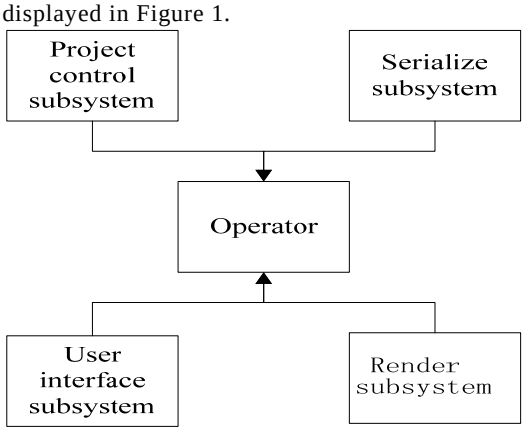


Fig. 1 The frame of simulation system

Since this simulation system simulates the 2D lamp model, we use the texture mapping of OpenGL to render the snapshots. The function void glTexImage2D(GLenum target,GLint level,GLint components,GLsizei width, glsizei height,GLint border,GLenum format,GLenum type, const GLvoid *pixels) defines a 2D texture mapping, where the parameter target is the constant GL_TEXTURE_2D. The parameter level represents the level of texture mapping of multilevel resolution. If there is only one resolution, then the level is defined as 0. The parameter component is an integer from 1 to 4 and it points out which of R, G, B, A can be chosen for adjusting and mingling. 1 represents choosing R, 2 represents choosing R and A, 3 represents choosing R, G, and B, and 4 represents choosing R, G, B, and A. The parameter width and height show the length and width of texture map. The parameter border is the width of boundary of texture and it often is 0. Width and height must be 2m+2b, where m is an integer, the length and width can be different value and b is the value of border. The maximum scale of texture mapping depends on OpenGL, but it at least use 64x64 (66x66 with the boundary). If the width and height is set to 0, then the texture will be closed effectively. The parameter format and type describe the format of the texture mapping and data type. It has the same meaning in the function glDrawPixels(). In fact, the texture data has the same format as the data used by glDrawPixels(). Part functions achieving the texture mapping in this system are displayed as bellow:

```
pTextureImg = LoadBMP(strEnvFileName))/* Input the BMP picture*/
glBindTexture(GL_TEXTURE_2D,m_pTextures[0]);/*Select the texture designed by m_pTextures[0]*/
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);/* Set the format of pixel*/
glTexImage2D(GL_TEXTURE_2D,0,GL_RGB,
pTextureImg->sizeX, pTextureImg->sizeY,0, GL_RGB, GL_UNSIGNED_BYTE, pTextureImg->data);/* Define the 2D texture function*/
glTexParameterf(GL_TEXTURE_2D,GL_TEXTURE_MIN_FILTER,GL_LINEAR);
glTexParameterf(GL_TEXTURE_2D,GL_TEXTURE_MAG_FILTER,GL_LINEAR); /*Filter wave*/
glTexEnvf(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE, GL_MODULATE);/*Pattern of mapping*/
gluBuild2DMipmaps(GL_TEXTURE_2D,3,pTextureImg->sizeX,pTextureImg->sizeY,GL_RGB,GL_UNSIGNED_BYTE, pTextureImg->data);/*Self-define the color and scale of texture map*/
```

4.2 Flow of simulation system

The commands flow of all the simulation system can be composed of the steps as follows:

1. Initialize the system; start to receive the user command, if there is no command, then wait;
2. Judge whether it is the command of controlling the lighting snapshot, if it is, then turn to step 3, else turn to step 1;
3. Send the command to the command object;
4. Check the current state from the property class and deal with it properly;
5. Submit the data dealt to the object of project class;
6. Inform the render system to change the display;
7. Change the state of the property class and then turn to step 1.

More details of flow see in Figure 2 displayed as bellow:

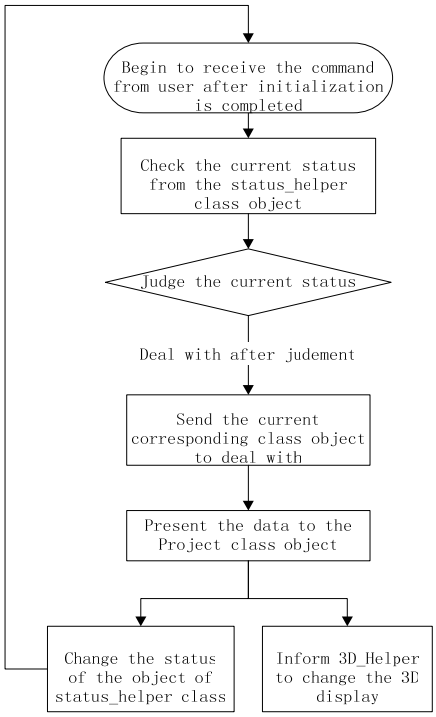


Fig. 2 The flowchart of system command

The simulation system inputs the configure file at the first time of initialization to control all kinds of maps of lamps. If the process of inputting is successful then it initializes the frames of MFC. It builds a new project file automatically after the main frame is formed. Meanwhile, it also initializes serialize of self-defined frames and the threads of play class. After the process of initialization, the system will wait for the change of state. If there is a change outside, for example, an operation of mouse, then judged what style this operation belong to by command class, and turn to the entrance of corresponding function to respond. If it is an operation of adding a lamp, then the system will judge that whether the operation is under the same unit. If it is an operation of adding a snapshot, then the system will judge that whether the operation is under the same lamp. If it is an operation of edit the property of snapshot, then the system will judge that whether the operation is under the same scene. All judgments are finished to display in the different dialogs. The play class displays the effects of messages from the command class on the main window by the message of project file. Therefore, one lamp or one snapshot is added and displayed on the user interface successfully. After adding kinds of lamps and snapshots constantly, the output class will form the output file. The output file has many different kinds according to certain protocol. The single player can play the file formed by the output class. After previewing the scene played on the computer, the system can connect to the SCM or other control systems and download the data by UART or FTP, and finally achieve the effect displayed on the computer. The effects of simulation in the daylight and in the night are displayed in Figure 3 and Figure 4.

Fig. 3 simulation effect in daylight

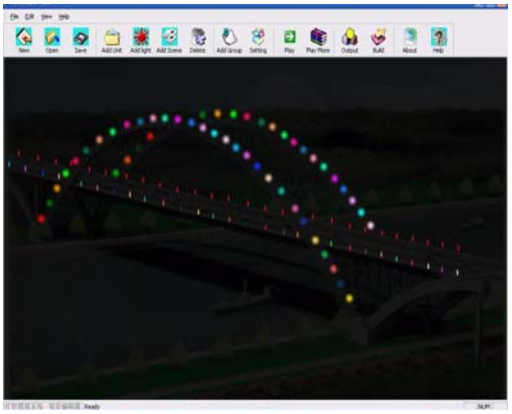


Fig. 4 simulation effect at night

The whole simulation system is composed of adding lamp model, adding snapshot of lamp, editing kinds of snapshots to merge a scene, and downloading the data. The system is developed based on object-oriented and the coupling degree among the modules is low but the independence is high. It provides the support to extend the system in the future.

5. CONCLUSION

There are few simulation systems which can simulate the scene of LED lighting. We design this simulation system to achieve the simulation of the scene composed of different snapshot of LED lighting based on the graphic library of OpenGL embedded in Visual C++. The simulation begins with adding kinds of lamp models, and then adding, editing, and previewing the scene, and finally downloading the data of scene to achieve the effect of simulation. The system makes the editing for the change of scene of LED lighting more convenient; meanwhile, it provides the necessary simulation for the LED lamps control systems

6. REFERENCES

[1] BERRY J K,BUCKL EY D J. Visualize realistic land2 scapes[J]. GIS World ,1998 ,11(8) :42 - 47.
[2] X.B. Zhou,S.S. Guo. lighting simulation based on OpenGL[J].Computer Simulation,2004, 21(1):93-95.
[3] David F Rogers. Computer graphics algorithm [M] . Beijing: China Machine Press,2002.
[4] Richard S. Wright,Jr. OpenGL super dictionary[M]. Beijing: Posts & Telecom Press,2001.
[5] Y.B. Bai,H.K. Shi. OpenGL 3D graphics programming guide[M] .Beijing: China Machine Press,1998.
[6] Y. Li. OpenGL implement [M] .Beijing: National Defense Industry Press ,2001.

On Image Fusion Using a Novel Swarm Intelligent Optimization Algorithm

Chunying Teng, Wenbo Xu
Institute of Information Technology, Southern Yangtze University
Wu Xi, Jiang Su, 214122,China
Email: tengchunying@hotmail.com

ABSTRACT

This paper proposes an image fusion approach based on QPSO algorithm. We formulate the image fusion problem as an optimization problem and adopt Quantum-behaved Particle Swarm Optimization algorithm to solve the problem. Not only QPSO has less parameter to control, but also does its sampling space at each iteration cover the whole solution space. Thus QPSO can find the best solution quickly and guarantee to be global convergent. In this paper, another two methods, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are tested for performance comparison with QPSO, and the result show the good efficiency of QPSO algorithms to image fusion.

Key words: QPSO Algorithm, Optimization, Pixel, Image Fusion.

1. INTRODUCTION

Image fusion is the technology that combines multi-information characters from different type sensor, using certain fusion algorithm to form new image. The new image is fused through visual information from multi-sensor. This fusion technique makes full use of the relativity on the space and the supplementary on the information of the original image. It makes a clearer and more full-scale expression and then strengthens the explanation ability of the image, improves the visual effect, and raises the dependability of image distinguish and the correction of the machine classification and identification. The image fusion technique is widely used, such as: medical science, remote sensing, military, and the computer vision and robot technique. It also plays an important part in image strengthen and feature pick-up.

According the expression method of the image, image fusion method can be classified into pixel level, feature level and decision level. In this paper pixel level is used. For pixel-level image fusion, Principle components analysis, evolution method and neural network method have been proposed. And the evolution method includes genetic method, ant system method, particle swarm optimization (PSO) and so on. These fusion methods can be distinguished through whether they are executed in space area or in transformation area. In this paper, a new PSO algorithm is proposed which can guarantee the global convergent: Quantum-behaved Particle Swarm Optimization algorithm. The experiment shows that the performance of QPSO is better than the other methods.

In this paper, the QPSO is introduced first, and then the basic image fusion method. In the end the image fusion method based on QPSO is proposed, and also do the comparison between PSO and GA, showing the results and conclusions.

2. QPSO ALGORITHM

QPSO is also a particle swarm optimization algorithm, using the conception of population and evolutionary and being operated according to the fitness value of its individual (particle). Every particle is considered as a particle that has no weight and size, flying in a certain speed in N_d dimensions search space. The flying speed is dynamically adjusted by the flying experience of every particle and that of the population. Each particle standing for a position in N_d dimensions space adjusts its place towards the following two directions:

The best position of each particle that has been found yet;

The best position of the population;

Every particle should include the following information:

$x_i = (x_{i1}, x_{i2}, ..., x_{id})$: The current position of every particle;

$v_i = (v_{i1}, v_{i2}, ..., v_{id})$: The current speed of every particle;

$P_i = (P_{i1}, P_{i2}, ..., P_{id})$: The best fitness value of the particle I ,that is pbest;

$P_g = (P_{g1}, P_{g2}, ..., P_{gd})$: The best fitness value of the population, that is gbest;

The evolution equation of the particle is:

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, ..., \frac{1}{M} P_{id}\right) \quad (1)$$

$$p_{id} = \varphi * P_{id} + (1-\varphi) * P_{gd} \qquad \varphi = rand \quad (2)$$

$$x_{id} = p_{id} \pm \alpha * \left| mbest_d - x_{id} \right| * \ln(\frac{1}{u}) \quad (3)$$

The $mbest$ here is the middle position of the $pbest$, p_{id} is the random value between P_{id} and P_{gd}. α is the shrink-expand coefficient of the QPSO. It is the important parameter that proves the convergence of QPSO. It is used as

α *=(1.0-0.5)*(MAXITER-T)/MAXITER+0.5*。

3. THE IMAGE FUSION METHOD BASED ON PIXEL-LEVELT

Suppose that there is an original white-black image R and its matrix is $R(m \times n)$. The image R is treated with Gauss noise and we get two images which matrixes are Ra and Rb. And the matrix Fr is fused by matrix Ra and Rb through linear authority:

$$Fr = \alpha \times Ra + (1-\alpha) \times Rb \qquad (4)$$

It can be seen that the fusion effect depends on α.

So, in order to consider the fusion effect, set fusion target G as:

$$G = \sqrt{\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [Fr(i,j) - R(i,j)]^2} \qquad (5)$$

The fitness function in image fusion can be calculated according to the Eq. (5). According to the Eq. (5), it is known that the smaller the fusion target G is, the smaller the error between fused image and original image is, and the better the fusion effect is.

4. THE PROCESS of THE IMAGE FUSION BASED ON QPSO

This algorithm is summarized as follows:

The pseudocode of the fusion method based on QPSO is:

For each particle

Initialization (each particle, local optimization, global optimization)

End

Do

For each particle calculate the matrix of the fused image according to the Eq. (4)

Calculate the fitness value according to equalized root (5)

If the particle's fitness value is smaller than the particle's best fitness value pBest

Set the current fitness value as the best fitness value pBest

End

Chose the smallest fitness value as the best particle, and set the fitness value as the global best fitness value gBest .

Calculate the mbest according to the Eq. (1)

For each particle

Calculate the random particle according to the Eq. (2)

Update the particle's best fitness value according to Eq. (3)

End

While meets the following two conditions do

1) Smaller than the biggest iterative number (the biggest iterative number is appointed by user);

2) After some iterative times, does not got the minimum error;

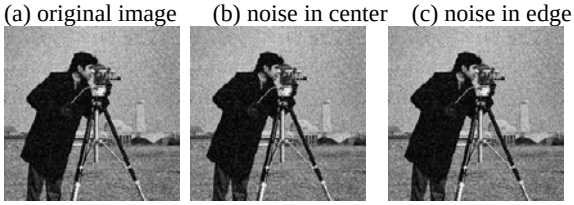
After the above approach, the best fusion image is gotten and the fusion method is finished.

5. THE RESULT AND CONCLUSION of THE EXPERIMENTT

There is an image（a）, about 226 × 227 that is used in this experiment, and another two images (b) and (c) which the center and edge of image (a)are treated with Gauss noise. Then the result is achieved from 100 times experiments of 20 particles and each experiment is made up of 50 iterative. And the result is compared with some other fusion methods: the genetic algorithm and the particle swarm optimization. And it is found that QPSO is better than them.

The followings are the experiment images and data:

(a) original image (b) noise in center (c) noise in edge



(d) GA (e) PSO (f) QPSO

Table 1. The average data of the experiment:

Algorithm	Square error		Average value	
GA	0.0012	0.00095	21.1129	21.1129
PSO	0.0015	0.0017	21.1127	21.1127
QPSO	8.5874e-004	8.9521e-004	21.1124	21.1124

The next two figures is the curve of the above three image fusion methods (we draw the first 50 times of the 100 iterative times):

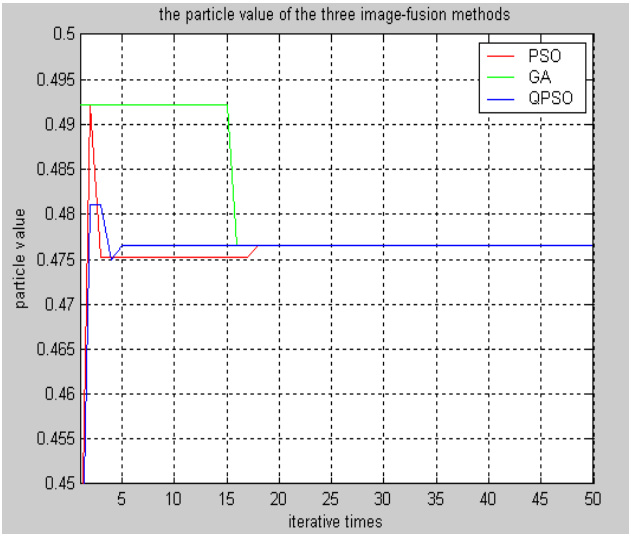


Fig. 1. The particle value of the three image-fusion methods

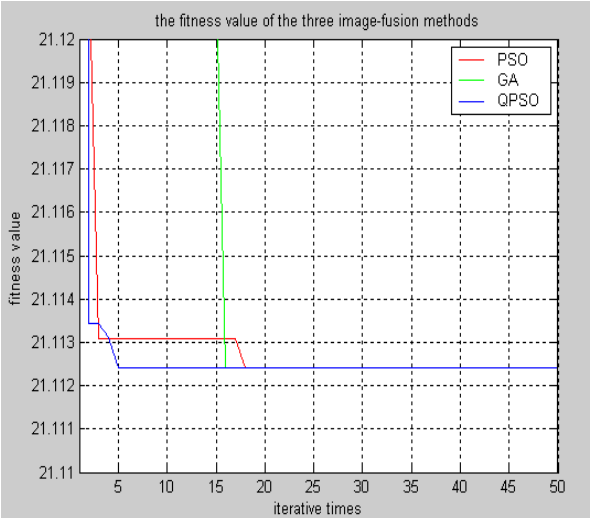


Fig. 2. The fitness value of the three image-fusion methods
（the blue stands for QPSO, the red stands for PSO, the green stands for GA）

We can find from the above data that the best fitness value can be found quickly (almost in the first 5 times) by using QPSO and from the square error and means we can know that QPSO is more steady than the other two methods. The GA and the PSO has many parameters to be treated and they are trouble to be realized. And compared with the above two methods, QPSO has less parameters and can be realized easily.

In addition, some other images are fused by this method proposed in this paper. And we get good effect. All fused results show that QPSO is steady and its fusion effect is satisfied.

6. REFERENCES

[1] J. Sun, W.B. Xu, “A Global Search Strategy of Quantum-behaved Particle Swarm Optimization”[C] Proceedings of IEEE conference on Cybernetics and Intelligent Systems, 2004,pp.111 – 116

[2] J. Sun, B.Feng, et al, ”Particle Swarm Optimization with Particles Having Quantum Behavior”[C]. Proceedings of 2004 Congress on Evolutionary Computation, 2004, pp.325-331

[3] R. Ramesh, l. Adrian, et al, “Image Fusion for Context Enhancement and Video Surrealism” the Association for Computing Machinery, 2004, pp.85-93.

Study on Deterministic Regularization Approach for Neuromagnetic Source Reconstruction

Jing Hu, Shiyan Ying
College of Information Engineering, Zhejiang University of Technology
Hangzhou, 310032, China
Email: hj@zjut.edu.cn

ABSTRACT

The Magnetoencephalography (MEG) inverse problem of reconstructing electrical sources in the human brain is indeed ill-posed and largely underdetermined. An efficient way of constraining the problem and thereby reducing the solution space is to perform regularization. Under the distributed source model, a region weighing approach present here is built on the minimum norm with Tikhonov regularization. One weighing matrix is considered from the algorithm feature of MEG equation, whereas another is from the sparse feature of source distribution to form an operator of region weighing so as to yield a satisfied solution. Computer simulations demonstrate this modified method is able to obtain a better source reconstruction.

Keywords: Ill-posed, Deterministic Regularization, Region Weighing, Reconstruction, Neuromagnetic Source.

1. INTRODUCTION

Magnetoencephalography (MEG), as the name suggests, measures the magnetic field outside the head. In an isolated environment, with properly tuned equipment, this field is due almost entirely to the inducted field resulting from the flow of current inside the head[1]. To address this physically ill-posed problem, constraints assumptions must be made about the neural source configuration. Although various reconstruction methods have been developed in research literature, in general, they may have a common structure summed up in a single word: regularization. Through regularization theory, the solution can thus be computed by optimizing suitable cost functional, which determines an implicit model for the MEG-based source imaging to find a unique, robust and physiologically justified solution.

2. THEORY AND METHOD

The reasonable solution for neuromagnetic source reconstruction may be obtained by various regularization techniques. In this paper, we adopt the weighted minimum norm estimation with Tikhonov regularization.

Tikhonov regularization [2] attempts to match the measured data while penalizing some function of the estimate to produce reasonable estimates. The degree of regularization is controlled by a parameter that governs the tradeoff between fidelity to the data and fidelity to the constraints. In order to obtain an acceptable solution, an appropriate regularization parameter λ must be chosen. If λ is too big, too much useful information could be lost. If λ too small, the numerical stability might be compromised.

Here, in order to get an analytical solution, the Euclidean norm is used as constraint. Let the data term be based on the least-squares criterion between the measured data and estimated data, so we have

$$L(b,p)=\|b-Gp\|_2^2 \tag{1}$$

Where b is the measured magnetic data, G is lead field matrix, and p is the reconstructed current density image.

And using a quadratic penalty term, the regularized criterion can be written in the form

$$R(p)=\|Wp\|_2^2 \tag{2}$$

Combine Eq.(1) and Eq. (2), the whole reconstruction scheme requires a strategy of choosing an appropriate λ such that the following cost function is satisfied and the norm of both data as well as prior term are small.

$$\|b-Gp\|_2^2+\lambda\|Wp\|_2^2 \tag{3}$$

A wide class of linear solutions can be constructed following this methodology. In general, different selections of W will produce different solutions. For examples, when W is equal to the identity matrix I, this is the standard Tikhonov regularization (also called the minimum norm estimation method). However, accordding to this method, the result has no physiological and anatomical importance. Namely, it tends to smear sources over the entire reconsturction region, containing nonzero values at virtually every grid node. Take account into the reconstructed current configuration must be consistent with physiological evidence which suggests that the neuromagnetic source image being of sparse and localized nature, i.e., the true dipole source is only located in a few small grid point and the current amplitude of the dipole source on the other grids is zero.

Here, we propose a region weighing operator to formulate a genenalized minimum norm estimation with Tikhonov regularization, and we call our method a region weighing method (RMN), it can be implemented by the following steps:

(i) obtain the minimum norm solution:

$$p_{mn}=(p_1,p_2,\cdots p_N) \tag{4}$$

where p_i corresponds to the current amplitude value on the ith grid. Let p_{mn} be the initial image.

(ii) Due to the decreasing sensitivity of the sensors, the minimum norm constraint introduces a bias towards superficial sources. To meet this limitation, it can be handled by the use of the lead field normalization method. So we choose the weighting matrix W_{depth} as a diagonal matrix in the transformed coordinate system

$$W_1=W_{depth}=\mathrm{diag}\left(\|G_i\|_2^{1/2}\right) \tag{5}$$

where G_i indicates the ith column of G.

(iii) Then we define a region weighing operator, i.e., set a diagonal weight matrix W_{region} is

$$W_2=W_{\mathrm{region}}=I-\mathrm{diag}\{\rho_i\} \tag{6}$$

Where $\rho_i=p_i^2/\sum_i p_i^2$

The reason lies in those dipoles having large current amplitude value would be enhanced and those having small amplitude value be suppressed.

(iv) Let $W = W_1 + W_2 = \mathrm{diag}\{\|G_i\|\} + (I - \mathrm{diag}\{\rho_i\})$, hence, the cost function becomes

$$\hat{p} = \arg\min_p \|b_{mea} - b_{cal}\|^2 + \gamma\|(W_1 + W_2)q\|^2 \quad (7)$$

3. SIMULATIONS

Assume the head model a spherical coordinate system with radius R = 10cm, centered in the origin. The sensors were positioned at the upper hemisphere close to the simulated source. 11cm from the center of sphere in the following manner: 1 sensor at the vertex, the other are positioned in rings around the sphere at 15 degrees intervals (until 90 degrees) with 4, 8, 12, 16 and 20 sensors on each ring. Thus, a simulated 61-channel SQUID magnetometer is used to record the MEG data for all experiments.

We apply the formula proposed by Cuffin[3], which is in the Cartesian coordinate system, so that the radial component of magnetic induction intensity can be yielded,

$$b_r = \frac{\mu\{(xy'-x'y)RP_\theta + [z'K_2 - z(x'x+y'y)]P_\phi\}}{4\pi R' K_1^{3/2} K_2^{1/2}}$$

$$K_1 = (x'-x)^2 + (y'-y)^2 + (z'-z)^2,$$

$$K_2 = x^2 + y^2$$

$$R = \sqrt{x^2+y^2+z^2}$$

$$R' = \sqrt{x'^2+y'^2+z'^2}$$

where (x,y,z) and (x',y',z') are the coordinates of source points and measured points respectively. P_θ and P_ϕ are two tangential components moment components of a dipole respectively (we don't need to consider the radical moment components since it does not contribute to the radical component of magnetic value in the spherical head model). Besides, $\mu = \mu_0 = 4\pi 10^{-7}\,\mathrm{Hm}^{-1}$.

Hence we get the magnetic induction intensity value outside the head for our simulation. Gaussian white noise is added to the experiments,

$$\mathrm{SNR} = 10\log(\textstyle\sum_{i=1}^{M} b_i^2 / \sum_{i=1}^{M} n_i^2).$$

Firstly, there are two dipoles, which locate in (2.0,2.0,7.0)cm and (-1.5,-1.5,7.0)cm in the brain to produce the outside magnetic field. Set the reconstructed source area be N grids, -4cm≤x≤4cm, -4cm≤y≤4cm, 5cm≤z≤9cm, point spacing is 5mm, so the total grid points N=2601. For simplicity, suppose the reconstructed sources are in the grid points, i.e., location parameters are not within the grid spacing. The dipole moments of two sources are 4.5nA·m and 1.5nA·m respectively. SNR is 5.4dB(28.8%noise). In order to reduce the noise interference, we perform 100 measurements, then get 100 sets of data, average it to increase the SNR, thus SNR is 10 times of the original one. See Fig.1, it is the magnetic field data of one measurement,

Take the assumed prior neurophysiologic information as a kind of weight. The region weighing will not affect the reconstruction results. Firstly compute the solution vector of standard minimum norm estimation so as to obtain the probability distribution of the reconstructed area, i.e., $\rho_i = p_i^2 / \sum_i p_i^2$. Give the interested area a big weighing, meanwhile the uninterested area a penalty, furthermore we part the local area of source gathering and the rest of sparse areas. Regularization parameter γ is adjusted by simulated experiment, here set it be 0.4. Fig.2 gives the reconstruction results, the top one is depth normalization weighing, below is the improved method. Results show the minimum norm

(vi) So the solution $\hat{p}$ is computed from

$$\hat{p} = G^+ b = (G^T G + \gamma(W_1 + W_2))^{-1} G^T b \quad (8)$$

estimation with W=W_1+W_2 is better than that only with W_1. The size of circles indicates the values of dipole moments. The direction of dipole can be determined by the three coordinate components of dipole moment, however in the real MEG experiment, the direction of dipole is known, vertical to the surface of cerebral cortex.

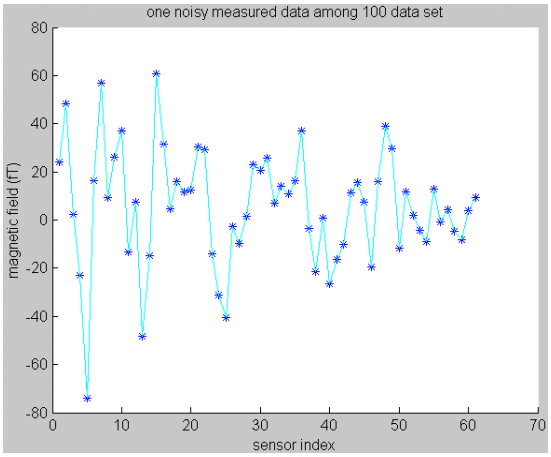


Fig. 1. Magnetic field measured noisy data b_{mea} in one measurement, the horizontal ordinate means the measured points outside the head, the vertical coordinate is the value of b_{mea}, totally 61 data, denoted by'*'.

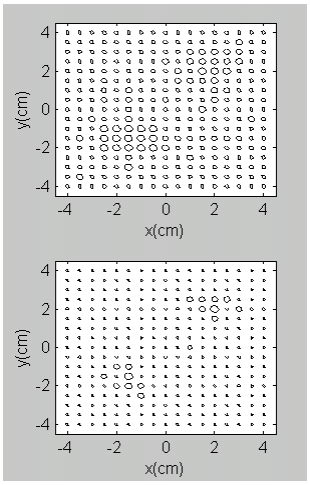


Fig. 2. Reconstruction results, top: W=W_1, bottom: W=W_1+W_2

Next, we do the comparison experiments between the depth-weight minimum norm estimation (WMN) and our region weight minimum norm (RMN). Assume the measured magnetic data outside the sphere are generated by three simulated dipole sources existing in the reconstructed region, i.e., three dipoles locating in (-5,-4,0),(5,-3,0),(-2,-7,0), and their moments being 15,10.8,1nAm respectively.

The additive Gaussian white noise is added into the experiment, SNR is 13dB (5%noise). An error parameter of reconstructed image is difined as

$$RE = \frac{(Gp-b)^T(Gp-b)}{b^T b} \cdot 100\% \quad (9)$$

The experimental results are, for WMN method, the value of *RE* is 6.012%, for RMN, *RE* turns to be 5.988%.

Furthermore, to describe the magnitude and direction of dipole sources, we draw three iso-magnetic field curve maps on the transverse, coronal and sagittal cross sections according to our adopted spherical coordinate system. Suppose every cross section tangent to the surface of spherical head, perpendicular to axis *x*, *y* and *z*, the crossing points is right auris point, nasion and vertex in the coordinate system.

Each section is set to be (-10, 10)×(-10,10) cm2. The MEG topographic map is shown in Fig.3. Since we only consider two tangent moment components of each simulated source, combine the location of three sources, three cross-sections can reflect clearly the dipole source situation close to the section, of course the direction has the deviation with our simulated sources, which just indicates the increase or decrease effect from other sources in the spherical head.

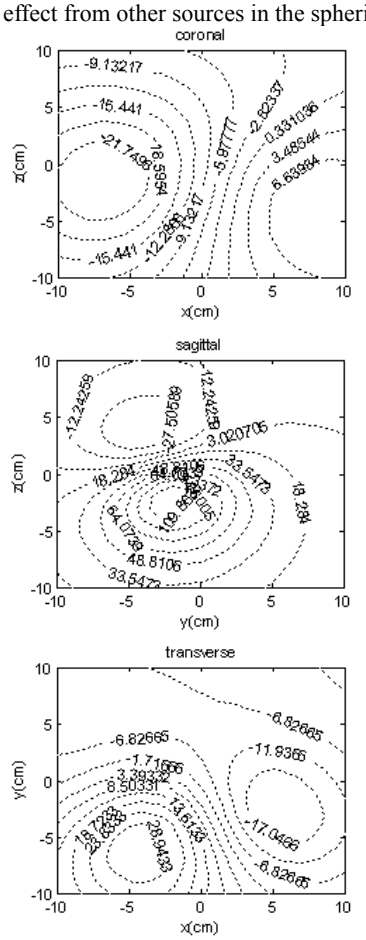


Fig. 3. Three topographic maps

4. CONCLUSION

Different cost functionals define different models. Here, the weighing matrix W_1 is considered from the algorithm feature of MEG equation, whereas W_2 is from the sparse feature of source distribution, the results indicates the method is effective.

However limitation exists in our method, that is, selection of prior probability distribution function is crucial, which decides the effectiveness of reconstruction algorithm. But in our experiments, it is only a formal selection for us to determine the probability distribution of dipole sources based on the minimum norm estimated solution, since we only want to show the feature of algorithm. If deviation of the initial solution is large, so that the next region weighing will be affected, then fail to perform reconstruction. Therefore the further work is needed.

5. ACKNOWLEDGEMENT

This work is supported by China postdoc science fund (2003033530) and Zhejiang provincial 14th startup fund for studying abroad and returning personnel.

6. REFERENCES

[1] J.M.Eklund, R.Bajcsy, "Computing MEG signal sources", Computations Systems Bioinformatics Conference, Stanford CA, Aug8-12, 2005

[2] A.N.Tikhonov, V.Y.Arsenin, Solutions of ill-posed problems. Winston, Wanshington,DC, 1977.

[3] B.N.Cuffin, "A comparison of moving dipole inverse solutions using EEG's and MEG's", IEEE Transactions on Biomedical Engineering, 32(11), 1985, 905-910.

A Kernel Fisher Discriminant Classifier for Speaker Recognition*

X. Li
Department of Electronic Science & Engineering, Nanjing University, Nanjing 210093, P. R. China
National Laboratory of Pattern Recognition, Institute of Automation of the Chinese Academy of Sciences, Beijing 100080, P. R. China
Email: xli@staff.shu.edu.cn
And
Y. Zheng
School of Computer Engineering and Science, Shanghai University, Shanghai 200072, P. R. China
Email: yuzheng@mail.shu.edu.cn

ABSTRACT

Feature extraction is the first important stage in an automatic speech recognition system. Traditional way to use different information sources is to concatenate different features into a long feature vector. Although it is simple to implement and works reasonably well, it has a few shortcomings. Firstly, not all the cepstral components contribute to the recognition accuracy. Secondly, the number of training vectors needed for robust density estimation increases exponentially with the dimensionality. In this paper, a kernel Fisher discriminant classifier was proposed. Instead of simply concatenating different sources features, we modeled each different feature set separately and made fusion of the KFD-based classifier output scores. A significant components selection procedure applying the kernel Fisher discriminant criterion, were presented to reduce the high dimensionality of the long feature vector before fed to the classifier. The speaker verification experiment indicates the superior recognition accuracy can be obtained in speech recognition.

Keywords: kernel Fisher discriminant, decision fusion, speaker verification, keywords spotting.

1. INTRODUCTION

Feature extraction transforms the raw speech signal into a compact representation that is more discriminative than the original signal. It is the first important stage in an automatic speech recognition system. Ideally, the feature space should be low-dimensional and spanned by independent features that have similar small intra-speaker variances and large inter-speaker variances for speaker recognition. That lead to compact, widely separated clusters for individual speaker [1]. In most automatic speaker and speech recognition systems, widespread used features are mel-frequency cepstral coefficients (MFCC) and linear predictive cepstral coefficients (LPCC) [2]. LPCC provides a good approximation to the vocal tract spectral envelope and suppresses harmonic information. However, the LP cepstrum is not robust to channel and noise effects. In contrast to pure linear predictive analysis of speech, Mel-scale cepstral analysis modifies the short-term spectrum of the speech by psychophysically based transformations. It has been found to be more robust to channel and noise. These features are often extended with the corresponding delta features which expected to carry information about vocal tract dynamics.

Traditional way to use these different information sources is to concatenate the cepstral vectors with their delta-cepstral (or delta-delta cepstral, if it is needed) into a long feature vector [3]. Although it is simple to implement and works reasonably well, it has a few shortcomings. Firstly, not all the cepstral components contribute to the recognition accuracy. Several cepstral components contain the most useful information about the speaker, while the others are useless or cause negative effects [4]. Secondly, the number of training vectors needed for robust density estimation increases exponentially with the dimensionality. This phenomenon is known as the curse of dimensionality [5].

Dimensional reduction is necessary for efficient and effective processing, especially for high dimensional data. The principal component analysis (PCA) and the linear discriminant analysis (LDA) are two of the most commonly used dimensional reduction methods. These methods search optimal directions for the projection of input data onto a lower dimensional space [6]. While the PCA finds the direction along which the data scatter is greatest, the LDA searches the direction which maximizes the inter-class scatter and minimizes the intra-class scatter. PCA is a useful tool for unsupervised and nonlinear problem for feature extraction. In the same manner LDA can be used for supervised and nonlinear problem for feature extraction and for classification [7].

Fisher Discriminant Analysis (FDA) [8], as one of the LDA, has been applied to feature transformation in many pattern classification problems. However, it has limitation for the data which are not linearly separable since it is difficult to capture a nonlinear relationship with a linear mapping. In order to overcome such a limitation, nonlinear extensions of it have been proposed.

Kernel Fisher Discriminant Analysis (KFDA) [9,10] tries to increase the expressiveness of the discriminant based on the high order statistics of the data set. Recently, Kernel Fisher Discriminant and some improved algorithms [11, 12, 13, 14, 15] have been applied to many pattern recognition problems and very promising results were reported.

However the particular drawback with the method is that the complexity will be increased with the number of training data. This implies that the testing time is going to be very slow for problems with a large number of training data and the computational cost for storing a large data set is also going to be high.

In this paper, data fusion are considered to alleviate this suffer without the increasing feature vectors. Most data fusion methods for speaker or speech recognition were concentrated to combine the scores from different models trained for a speaker or a speech [16, 17, 18]. It depends on the performance of the classifiers but the features fed to the classifiers are derived from the same source. An alternative to the classifier-based fusion is to model each different feature set separately, to design a specialized classifier for

this feature set, and combine the classifier output scores or the decisions together [19]. Therefore, a framework combines the kernel Fisher discriminant criterion and decision fusion was proposed. Instead of simply concatenating different sources features, we modeled each different feature set separately and made fusion of the classifier output scores. A significant components selection procedure applying the kernel Fisher discriminant criterion was presented to reduce the high dimensionality of the long feature vector before fed to the classifier.

2. SIGNIFICANT COMPONENTS SELECTION BY KERNEL FISHER DISCRIMINANT CRITERION

2.1. Linear Fisher discriminant (LFD)

LFD is a mathematical transform algorithm from *d*-dimensional input space to one-dimensional output space. Given a set of *n* *d*-dimensional samples $\{x_1, x_2, \cdots, x_n, x_i \in \Re^d\}$ with n_1 samples in class 1 denoted C_1 and n_2 samples in class 2 denoted C_2 then the Fisher linear discriminant [6, 20] is given by the vector $\mathbf{w}\left(w \in \Re^d\right)$, which maximizes the following function

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{1}$$

where S_B and S_W are respectively the between-class and the within-class scatter matrices defined as follows,

$$S_B = (m_1 - m_2)(m_1 - m_2)^T \tag{2}$$

$$S_W = \sum_{i=1,2} \sum_{x \in C_i} (x - m_i)(x - m_i)^T \tag{3}$$

m_i is the sample mean of the respective classes defined as

$$m_i = \frac{1}{n_i} \sum_{x \in C_i} x \tag{4}$$

w which maximizes the Eq. (1) is regarded as the optimal projection direction, shown by

$$w = S_W^{-1}(m_1 - m_2) \tag{5}$$

Projection of a testing sample x onto the direction w is $G(x) = w^T x$, we can decide which class sample x belongs to by the decision function $F(x) = \text{sign}(G(x) - w_0)$, where w_0 represents decision threshold.

2.2. Kernel function Fisher discriminant (KFD)

Let Φ be a nonlinear mapping from the input feature space $\Re^d$ into the implicit higher dimensional (maybe infinite) feature space *F*. Then, we can find Fisher's discriminant in *F* by maximizing

$$J(w) = \frac{w^T S_B^\Phi w}{w^T S_W^\Phi w} \tag{6}$$

where w is a column vector of W in *F*, S_B^Φ and S_W^Φ are respectively the between-class and the within-class scatter matrices in *F*,

$$S_B^\Phi = (m_1^\Phi - m_2^\Phi)(m_1^\Phi - m_2^\Phi)^T \tag{7}$$

$$S_W^\Phi = \sum_{i=1,2} \sum_{x \in C_i} (\Phi(x) - m_i^\Phi)(\Phi(x) - m_i^\Phi)^T \tag{8}$$

Here,

$$m_i = \frac{1}{n_i} \sum_{x \in C_i} \Phi(x) \tag{9}$$

Since $\Phi(x)$ is impossible to be computed directly, a kernel function $k(x_i, x_j) = \Phi(x_i)^T \bullet \Phi(x_j)$ is introduced, which satisfies the Mercer's condition [21]. From the theory of reproducing kernels, we know that any solution $x \in F$ must lie in the span of all data samples in *F*, i.e.,

$$w = \sum_{i=1}^{n} \alpha_i \Phi(x_i) \tag{10}$$

Let $\alpha^T = [\alpha_1, \cdots, \alpha_n]_{1 \times n}$, we can find Fisher's discriminant in *F* by maximizing

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \tag{11}$$

where $M = (M_1 - M_2)(M_1 - M_2)^T$ with $(M_i)_j = \frac{1}{n_i} \sum_{x \in C_i} k(x, x_j)$ and $N = \sum_{i=1,2} K_i (I - 1_{n_i}) K_i^T$.

K_i is a $n \times n_i$ matrix with element $(K_i)_{a,b} = k(x_a, x_b), x_a, x_b \in C_i$. I am the identity matrix and 1_{n_i} am the matrix with all elements set to $1/n_i$.

Eq. (11) can be solved by finding the leading eigenvector of $N^{-1}M$ and the projection of a new sample x onto w is given by

$$\Phi(x)^T w = \sum_{i=1}^{n} \alpha_i k(x, x_i) \tag{12}$$

As a result, the problem for obtaining w is transformed into one for solving eigenvectors α, which corresponds to the maximum J. The optimal J will be solved by the eigenequation

$$M\alpha = \lambda N\alpha \tag{13}$$

So the eigenvector, corresponding to the maximum eigenvalue of Eq. (13), is the optimal α. In fact, for two-class classification the optimal α can be achieved by

$$\alpha = N^{-1}(M_1 - M_2) \tag{14}$$

In practice, generally N is singular, so α can be solved by the following equation:

$$\alpha = (N + \mu I)^{-1}(M_1 - M_2) \tag{15}$$

where μ is a positive constant. The Fisher criterion can be defined as

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T (N + \mu I)^{-1} \alpha} \tag{16}$$

2.3. Significant components selection

Following [11, 12], a linear combination of "significant nodes" to approximate discriminant vector in feature space can be decided. These "significant nodes" can also be taken as the most important feature components which could be used for feeding to the classifier.

From Eq. (16), it is provable [11] that

$$J(\alpha) = (M_1 - M_2)^T \alpha \tag{17}$$

Obviously, the greater $(M_1 - M_2)^T \alpha$ is, the more significant the corresponding patterns are. So $(M_1 - M_2)^T \alpha$ can be taken as a criterion to select the significant components of the features. The detailed

procedures are described in [11, 12]. For a test pattern x_t, $f(x_t)$ can be obtained by

$$f(x_t) = \sum_{i=1}^{r} \alpha_i k(x_t, x_i^o) \qquad (18)$$

Here, r is the number of the significant components, and x_i^o is the selected significant components.

3. FUSION STRATEGY AND THE STRUCTURE OF THE SYSTEM

We use different feature sets and combine their output scores from the classifier. Ref. [22] compared several commonly used fusion criteria and indicated that the sum rule is most resilient to estimation errors. Therefore, we define the combination rule as the weighted sum:

$$S_{total} = \sum_{j=1}^{M} w_j s_j \qquad (19)$$

Where M is the number of feature sets and w_j is the weight for the feature set. The weights are normalized such that $\sum_{j=1}^{M} w_j = 1$, which allows the weights to be interpreted as relative importance. Since for different purpose, w_j should be dissimilitude. So we apply a BP neural network to ascertain the value of w_j. The proposed fusion structure of the system is shown in Fig. 1.

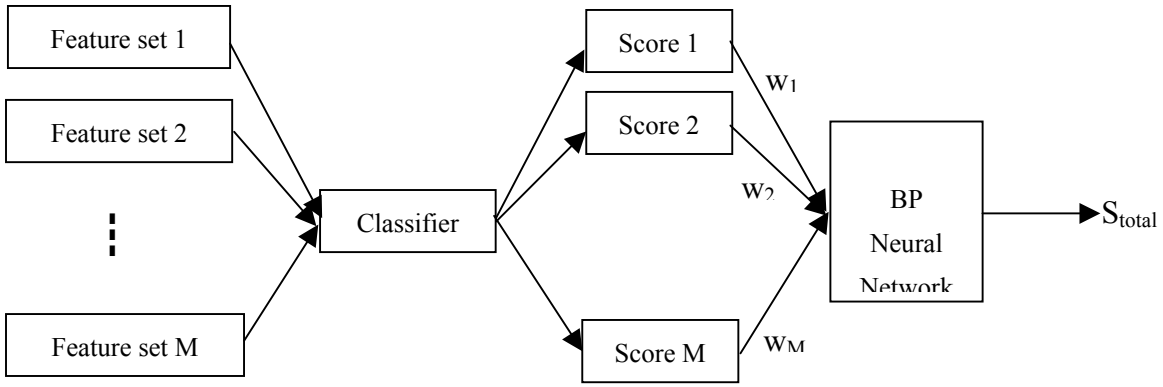


Fig. 1. Structure of the proposed system

4. EXPERIMENT

In the following experiment, elliptical basis function (EBF) networks [23] were used to construct audio models for contrasting with KFD method we introduced before. An *n*-input, *m*-output EBF network can be considered as realizing a multi-dimensional non-linear transform that maps data from R^n to R^m. The *k*-th output $(k = 1,\ldots,K)$ of an EBF network transforms an input vector $\vec{x}_p$ into

$$w_{k0} + \sum_{j=1}^{J} w_{kj} \exp\left\{-\frac{1}{2\gamma_j}(\vec{x}_p - \vec{\mu}_j)^T \Sigma_j^{-1}(\vec{x}_p - \vec{\mu}_j)\right\} = y_k(\vec{x}_p) \approx f(\vec{x})$$

$$p = 1,\ldots,N \qquad k = 1,\ldots,K \qquad (20)$$

where $\vec{\mu}_j$ and Σ_j are the mean vector and covariance matrix of the *j*-th basis function respectively, w_{k0} is a bias term, w_{kj} is an output weight connecting the hidden node ***j*** to the output node *k*, and γ_j is a smoothing parameter that controls the spread of the ***j***-th basis function. It was determined heuristically by

$$\gamma_j = \alpha \sum_{l=1}^{L} \left\| \vec{\mu}_l - \vec{\mu}_j \right\| \qquad j = 1,\ldots,J \qquad (21)$$

where $\vec{\mu}_l$ denotes the *l*-th nearest neighbor of $\vec{\mu}_j$ in the Euclidean sense, *L* is the number of nearest neighbors, and α is a parameter that controls the spread of the basis functions. The reason we applying the EBF networks is, it generally achieve better performance than the standard RBF networks for pattern recognition [24].

To show the possible improvement in performance of the proposed KFD-based classifier compared with the EBF-based classifier, speaker verification experiment was examined.

Speaker verification is the verification of whether the voice of a claimant matches the voice of the claimed identity. We used 190 speakers from the TIMIT and NTIMIT corpuses to evaluate the performance. TIMIT is a phonetically balanced, continuous speech corpus, and NTIMIT was obtained by playing the speech in the TIMIT corpus through a telephone network, which resulted in a telephone bandwidth corpus [25]. Each speaker in the speaker set (76 speakers from dialect region 2) was assigned a personalized EBF network or a KFD discriminant classifier that modeled the characteristics of his/her own voice. The anti-speakers and impostors are randomly selected from dialect regions 1 (38 speakers) and 3 (76 speakers).

Only spectral features we used, 12-th order LPCCs and MFCCs with their delta features were respectively computed every 14ms using a Hamming window of 28ms. Instead of simply concatenating these features, we modeled each different feature set separately and made fusion of the classifier output scores. We built speaker models using different feature sets with 24 or 48 dimensions each. LPCC and MFCC features are standard 12 cepstra with their dynamic features respectively. KFD features are obtained by

transforming 84 dimensional LFCC or MFCC features from 7 neighboring frames to 24 dimensions and applying for the classifier. Gaussian kernel in the form of $k(x,y)=\exp\left(-\|x-y\|^2/(2\sigma^2)\right)$ is adopted for KFD analysis, where the parameters σ and μ of Eq. (14) are set empirically, which is 0.0001 and 0.001 respectively in this experiment.

Table 1. shows the false accept rate (FAR) and the false rejection rate (FRR) based on the average of 76 speakers and 76 imposters. The average error rate (AVE) is the square root of the product of FAR and FRR. Here, LPCC+MFCC means concatenating of MFCC with LPCC and their delta derivations, and LPCC*MFCC means feeding them to the classifier respectively then making decision fusion of classifier's outputs.

The experiment results show applying KFD analysis will achieve better performance than using EBF-based classifier. From the table, it can be seen that the proposed structure generally perform best than the others.

As it can be observed in the table, applying different feature sets fed to classifier for decision fusion was helpful to reduce the recognition error rate. In our study, fusion-based methods yield reliable recognition results by fusing average matching score.

Table 1. Test error rate for TIMIT and NTIMIT

Corpus		TIMIT			NTIMIT		
classifier	Feature set	FAR%	FRR%	AVE%	FAR	FRR	AVE%
EBF-based classifier	LPCC+ΔLPCC	0.11	2.56	0.53	11.56	14.66	13.02
	MFCC+ΔMFCC	0.12	2.04	0.49	10.22	13.36	11.82
	LPCC+MFCC	0.10	1.64	0.40	11.40	14.02	12.64
	LPCC*MFCC	0.08	1.92	0.39	9.24	12.86	10.90
KFD-based classifier	LPCC+ΔLPCC	0.12	2.36	0.53	9.88	13.42	11.51
	MFCC+ΔMFCC	0.06	2.40	0.38	9.24	13.14	11.02
	LPCC+MFCC	0.04	1.87	0.27	8.99	12.48	10.59
	LPCC*MFCC	0.02	2.02	0.20	9.06	11.87	10.37

5. CONCLUSION

This paper presents and compares KFD-based and EBF-base classifier for speaker and speech recognition. Significant components selection and data fusion produces are integrated in the framework. The experiment results show that KFD-based algorithm could achieve equivalent or improve the recognition accuracy. And Applying decision fusion with average matching scores also demonstrated superior recognition accuracy. It is helpful for problems with a large number of training data and reduces the computational cost for storing a large data set.

6. REFERENCES

[1] D. O'Shaughnessy, Speaker Recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 3:4-17, 1986.

[2] X. Huang, A. Acero et al, Spoken Language Processing: a Guide to Theory, Algorithm, and System Development. Prentice-Hall, New Jersey, 2001.

[3] J. Campbell. Speaker Recognition: A Tutorial, Proceedings of the IEEE, 1997,85(9):1437-1462.

[4] B. Zen, X. Wu, et al, On the Importance of Components of the MFCC in Speech and Speaker Recognition. ACTA SCIENTIARUM NATURALIUM UNIVERSITATIS PEKINENSIS, , 2001 No. 3.

[5] C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, New York, 1996.

[6] R.O. Duda, P.E. Hart et al, Pattern Classification. Wiley-interscience, New York, 2001.

[7] G. Baudat, F. Anouar. Generalized Discriminant Analysis Using a Kernel Approach. Neural Computation, Oct. 2000, 12(10): 2385-2404.

[8] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd. ed., Academic Press, 1990.

[9] S. Mika, G. Rätsch, et al, Fisher Discriminant Analysis with Kernels. Neural Networks for Signal Processing, 1999, 5:41-48.

[10] M. Xu, X. Wu, et al, Context Quantization by Kernel Fisher Discriminant. IEEE Transactions on Image Processing, 2006, 15(1): 169-177.

[11] Y. Xu, J. Y. Yang et al, A Reformative Kernel Fisher Discriminant Analysis. Pattern Recognition, June 2004, 37(6): 1299-1302.

[12] Y. Xu, J. Y Yang, et al, An Efficient Renovation on Kernel Fisher Discriminant Analysis and Face Recognition Experiments. Patter Recognition, Oct., 2004, 37(10): 2091-2094.

[13] S. A. Billings, K.L. Lee. Nonlinear Fisher Discriminant Analysis Using a Minimum Squared Error Cost Function and the Orthogonal Least Squares Algorithm, Neural Networks, 2002, 15(2): 263–270.

[14] C. H. Park, H. Park, Nonlinear Feature Extraction Based on Centroids and Kernel Functions. Patter Recognition, April 2004, 37(4): 801-810.

[15] Y.H. Chao, H.M. Wang, et al, Gmm-Based Bhattacharyya Kernel Fisher Discriminant Analysis for Speaker Recognition. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05) , March, 2005, 1:649 – 652.

[16] T. Kinnunen, V. Hautamäki et al, On the Fusion of Dissimilarity-Based Classifiers for Speaker Identification. Proc. 8th European Conference on Speech Communiation and Technology (Eurospeech 2003) , September, 2003, 2641-2644.

[17] R. Ramachandran, K. Farrell et al, Speaker Recognition - General Classifier Approaches and Data Fusion Methods. Pattern Recognition, 35: 2801—2821, 2002.

[18] K.R. Farrell, R.P. Ramachandran, et al, An Analysis of Data Fusion Methods for Speaker Verification. 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '98), May 1998, 2:1129-1132.

[19] T. Kinnunen, V. Hautamäki, et al, Fusion of Spectral Feature Sets for Accurate Speaker Identification. Proc. 9th International Conference Speech and Computer (SPECOM'2004) , September, 2004, 361-365.
[20] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, 1996.
[21] C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 1998, 2:121-167.
[22] J. Kittler, M. Hatef, et al. On Combining Classifiers. IEEE Trans. Pattern Analysis and Machine Intelligence, 1998, 20(3): 226-239.
[23] X. Li, Y. Zheng, et al,Feature Mapping and Recuperation by Using Elliptical Basis Function Networks for Robust Speaker Verification. Journal of Shanghai University, 2002, 6(4): 331-336.
[24] M.W. Mak, C. K. Li, et al, Maximum Likelihood Estimation of Elliptical Basis Function Parameters with Application to Speaker Verification. International Conference on Signal Processing (ICSP'98) , Oct., 1998, 1287-1290.
[25] C. Jankowski, A. Kalyanswamy, et al, NTIMIT: A Phonetically Balanced, Continuous Speech Telephone Bandwidth Speech Database. Proc. International Conf. Acoustics, Speech and Signal Processing (ICASSP'90) , 1990, 109-112.

A Speech Recognition Approach with MFCC and Fractal Dimension

Minghai Yao, Jing Hu
College of Information Engineering, Zhejiang University of Technology
Hangzhou, 310032, China
Email: ymh@zjut.edu.cn

ABSTRACT

A modified approach present here combines Mel-frequency cepstral coefficients (MFCC) and fractal dimension as mixed feature parameter to carry out the speech recognition. Due to the respective advantages on expressing speech signal of MFCC and fractal dimension, fractal dimension denotes the self-similarity, periodicity and randomness of speech signal; meanwhile MFCC feature parameter describes speech nonlinearity. Besides, BP neural network is introduced into the whole speech recognition training procedure. Experimental results demonstrate fractal dimension is able to reflect the speech feature to some extent, makeup the shortcoming of traditional speech feature. The recognition performance is then improved by introducing fractal dimension feature.

Keywords: Speech Recognition, MFCC, Fractal Dimension, neural network, mixed parameter

1. INTRODUCTION

Speech recognition attracts much attention due to its broad application. The key point is speech feature selection and extraction. So far, there are many literatures about studying speech feature parameters, such as critical-band feature vector, linear spectrum pair, linear prediction coefficients etc.

The most applicable ones are linear predictive cepstral coefficients (LPCC) and Mel-Frequency cepstral coefficients (MFCC) based on the sound channel model and auditory mechanics. MFCC, the cepstral parameters extracted from the Mel-scale frequency domain, describes the nonlinearity of the ear's perception on frequency. Because of having the higher spectrum resolution at low frequency segment and better noise robust than LPCC, MFCC gets wide applications recently.

However, speech signal is a complex nonlinear procedure, i.e., according to the acoustics and aerodynamics, it includes the glottis nonlinear oscillation procedure changed by tongue and shape of sound channel, but also might produce eddy current on the border layer of sound channel especially for those friction and burst sound etc., which would finally form the turbulent current, while uttering other sounds, the turbulent current still exists in the air current from glottis, and the turbulent current itself is just a kind of chaos.

That is to say speech signal consists of chaotic natural phoneme, where exists the chaos mechanics [1, 2]. It is hard to improve the performance of traditional speech recognition technique built on the linear system theory, so researchers then turn to apply nonlinear system theory [3].

Here, by chaos characteristics of speech signal, speech fractal dimension as one feature parameter in recognition to combine with the traditional linear feature parameter MFCC, we present a speech recognition method with mixed feature parameter to improve the recognition performance.

2. THEORY AND METHOD

2.1 Speech Fractal Dimension Computation

Fractal dimension is a quantitative value from the scale relation on the meaning of fractal, and also a measuring on self-similarity of its structure. The fractal measuring is fractal dimension. From the viewpoint of measuring, fractal dimension is extended from integer to fraction, breaking the limit of the general topology set dimension being integer. Fractal dimension, fraction mostly, is dimension extension in Euclidean geometry.

There are many definitions on fractal dimension, e.g., similar dimension, Hausdoff dimension, information dimension, correlation dimension, capability dimension, box-counting dimension etc., where, Hausdoff dimension is oldest and also most important, for any sets, it is defined as [4]:

$$D = \lim_{\delta \to 0} (\ln M_\delta(F) / \ln \delta^{-1}) \quad (1)$$

Where, $M_\delta(F)$ denotes how many unit δ needed to cover subset *F*.

The definition of Hausdorff dimension is kind of abstract, i.e., it is not easy to obtain the speech fractal dimension directly. It needs to get embodied, i.e., the phoneme signal taken as an example, speech signal is knows as a 2-D space subset *F*, such that the measure value $M_\delta(F)$ may be determined by the step amount of using the space distance δ to measure the entire *F*.

Generally, $M_\delta(F)$ subjects to the power function. For constants *c* and *s*, while $\delta \to 0$, so that $M_\delta(F) = c\delta^{-s}$, then apply the logarithm to both sides to get,

$$\ln M_\delta(F) \approx \ln c - s \ln \delta \quad (2)$$

Where the difference is approaching zero as δ approaches zero, such that

$$s = \lim_{\delta \to 0} [\ln M_\delta(F) - \ln \delta^{-1}] \quad (3)$$

s can be estimated by the slope of logarithm figure by the value of δ in an appropriate range.

Experiments proved, when $\delta \to 0$, the slope of $\ln M_\delta(F) - \ln \delta^{-1}$ will approach gradually to a certain value *s*, and this slope *s* is just the fractal dimension in 2-D space subset *F*.

Sampled speech signal is not continuous, thus it is unfeasible to assume $\delta \to 0$. In literatures, interpolation on the sampled signal was tried based on fractal theory. However, signal interpolation might be unreal, moreover, while δ is little, slope has tended to a certain value, therefore needless to set a very little δ.

In the algorithm [5], the sampling rate is 8 kHz, 16-bit sampling on each sample data. Furthermore, segment is run by the regular window along the speech wave, and the fractal dimension on speech wave in each window is then

calculated. Each window is divided orderly into *r* mean + segments,

$$r = 1/\delta_i = 2^i, i = 1,2,3,\cdots,n,$$

in the j*th* segment among *r* segments (from speech wave[k] to speech wave[1], we have

$$(M_{\delta_i}(F))_j = sqrt((speechwave[l] - speechwave[k])^2 + (l-k)^2), \quad (4)$$

Such that $M_{\delta i}(F) = \sum_j (M_{\delta i}(F))_j$.

Then apply

$$D = \lim_{\delta i \to 0} (\ln M_{\delta i}(F) / \ln \delta_i^{-1})(i = 1,2,3,\cdots,n) \quad (5)$$

to fit *n* points, the slope we obtain is the fractal dimension *D*, namely,

$$fractal(speechwave[point]) = \frac{n\sum_{i=1}^{n} \ln M_{\delta i}(F) \ln 2^i - \sum_{i=1}^{n} \ln 2^i \sum_{i=1}^{n} \ln M_{\delta i}(F)}{n\sum_{i=1}^{n} (\ln 2^i \ln 2^i) - \sum_{i=1}^{n} \ln 2^i \sum_{i=1}^{n} \ln 2^i} \quad (6)$$

2.2 MFCC Computation

Auditory is a special nonlinear system, which has various sensitivity responding speech signals in the various frequencies, so it basically is a logarithm relation. MFCC is a kind of parameter making good use of ear's special perceptional nature. Lots of studies show, MFCC can improve the system recognition performance more than any other parameters like LPCC.

"Bark" as the base frequency in MFCC computation, the transformation with the linear frequency can be written by,

$$f_{mel} = 2595 \log_{10}(1 + f/700) \quad (7)$$

In (7) parameter can be obtained by frames. First, we adopt FFT to get the power spectrum $S(n)$ of frame signal, which is then transferred to be the power spectrum in Mel frequency. Before computation, it is necessary to set several band filters within signal frequency spectrum range.

$$H_m(n), \quad m = 0,1,\cdots,M-1, \quad n = 0,1,\cdots,N/2-1$$

Where *M* is the number of filters, generally 24; *N* is the number of points in one frame signal.

Each filter is a simple triangle at frequency domain, and its central frequency is f_m, even distributed on Mel frequency axis. At linear frequency, while m is little, the distance of adjacent f_m is very small, when m increases, the distance of adjacent f_m will increase gradually. The parameters of band filters should be evaluated in advance, then directly to be used to yield MFCC parameters. Generally, by the following steps, MFCC parameter can be obtained.

1): Firstly, set the point number of each frame sampled sequence, here, N=240 points. Do the pre-emphasis processing on each frame sequence, and then by the discrete FFT (DFFT), we have the discrete power spectrum $S(n)$, i.e., the square of mode.

2): After passing by M $H_m(n)$, we get the power value of $S(n)$, i.e., computing the product sum of $S(n)$ and $H_m(n)$ at each discrete frequent point, we then obtain M parameters $P_m, m = 0,1,\cdots,M-1$.

3): Get $L_m, m = 0,1,\cdots,M-1$ from the natural logarithm of P_m.

4): For $L_0, L_1, \cdots, L_{m-1}$, evaluate their discrete cosine transformation (DCT), say, $D_m, m = 0,1,\cdots,M-1$.

5): Discard the direct current item D_0, name $D_1, D_2, \cdots, D_K$ as MFCC parameter, *K*=12.

The computation flow chart is shown in Fig.1.

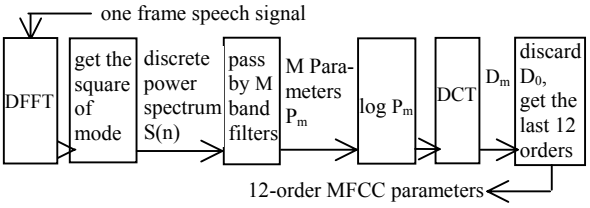


Fig. 1. The flow diagram of MFCC parameter computation

2.3 Recognition Method with Mixed Parameters

Considering the respective advantages on expressing speech signal of MFCC and fractal dimension, we mix both to be the feature signal, that is, fractal dimension denotes the self-similarity, periodicity and randomness of speech time wave shape, meanwhile MFCC feature parameter describes speech nonlinearity.

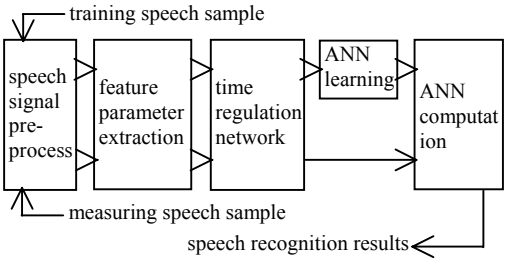


Fig. 2. ANN-based speech recognition procedure

Due to ANN's nonlinearity, self-adaptability, robust and self-learning such obvious advantages, its good classification and input-output reflection ability are suitable to resolve speech recognition problem. Fig.2 gives the whole training recognition procedure based on BP neural network.

3. EXPERIMENTS

To investigate this recognition approach proposed in this paper, ten Chinese digits from 0 to 9 are used in our experiments.

The sampling rate is 8 kHz; precision ratio is 16-bit, frame length 256, frame shift 64. A series of preprocessing are included, e.g., speech signal pre-emphasis, framing processing, windowing processing and endpoints detection etc., MFCC parameter of Chinese digit speech is then extracted, next, the algorithm above is used to evaluate MFCC parameter and fractal dimension for each frame speech.

Here, The Chinese digit speech was recorded in our lab. Speakers include three males and two females. Everyone uttered each word for ten times, totally 500 speech samples.

The training set is made from the first five utterances of everyone, i.e., totally 250 samples, the remaining 250 samples are grouped as the testing set.

Due to the number of ANN input nodes being fixed, therefore time regularization is carried out to the feature parameter before inputted to the neural network.

In our experiments, MFCC and fractal dimension of each sample are need to get through the network of time regularization separately[6]， MFCC is 4-frame data ($MFCC_1$-$MFCC_4$, each frame parameter is 12-D)，fractal dimension is regularized to be 12-frame data (F_1,F_2,…,F_{12}, each frame parameter is 1-D)，so that the feature vector of each sample has $4\times12+12\times1$=60-D，the order is, the first 48 dimensions are MFCC, the rest 12 dimensions are fractal dimensions.

Thus, such mixed feature parameter can show speech linear and nonlinear characteristics as well. By a large number of experiments, we build two different structures of BP neural network. One is used for mixed feature parameter method, the layers of input; hidden and output are 60, 24 and 10 respectively. Another is for single feature parameter method (MFCC only), 48 input, 24 hidden and 10 output layers included.

Two schemes are given for two different feature parameter methods.

1) Only extract MFCC feature, the feature vector Y_m of the input neural network is 48-D,

$$Y_m = MFCC_1 + MFCC_2 + MFCC_3 + MFCC_4 \quad (8)$$

2) Extract the mixed feature of MFCC and fractal dimension, then Y_m is 60-D,

$$Y_m = MFCC_1 + MFCC_2 + MFCC_3 + MFCC_4 + F_1 + F_2 + ... + F_{12} \quad (9)$$

Tab.1-3 gives the experiments results.

Our experiments show the recognition performance by mixed parameter of MFCC and fractal dimension is better than that by single MFCC parameter. Therefore, the recognition ratio is improved by adding fractal dimension to some extent. Besides, we cannot get the ideal results if only using fractal dimension as feature parameter. That means fractal dimension is not the main speech feature, which need to combine the traditional feature parameter to obtain the good recognition results.

Table 1. Recognition results by two schemes

Feature parameter Type	Sample number	correct recognition number	correct recognition ratio (%)
Single MFCC parameter	250	213	85.2
Mixed parameter with MFCC& fractal dimension	250	219	87.6

Table 2. Recognition results with MFCC parameter only

Chinese digit speech	0	1	2	3	4	5	6	7	8	9
Measure sample number	25	25	25	25	25	25	25	25	25	25
Correct recognition number	21	20	23	22	22	24	17	21	22	21
Correct recognition ratio (%)	84	80	92	88	88	96	68	84	88	84

Table 3. Recognition results with the mixed parameter

Chinese digit speech	0	1	2	3	4	5	6	7	8	9
Measure sample number	25	25	25	25	25	25	25	25	25	25
Correct recognition number	20	22	22	23	23	21	20	22	23	23
Correct recognition ratio (%)	80	88	88	92	92	84	80	88	92	92

4. CONCLUSION

We propose a speech recognition approach with mixed parameter in this paper, which combines the traditional MFCC and fractal feature as the feature parameter. MFCC has higher spectrum resolution at low frequency segment, while it cannot represent speech nonlinearity. Fractal dimension is used to quantitatively describe the chaos nonlinearity in speech airflow. Experimental results demonstrate this method is promising in improving speech recognition performance.

5. REFERENCES

[1] P. Maragos, “Fractal aspects of speech signals: dimension and interpolation”, Proc. IEEE ICASSP, 1991, 417-420.

[2] S. Sabanal, M. Nakagawa, “The fractal properties of vocal sounds and their application in the speech recognition model”, Chaos, Solitons & Fractals, 1996, 7(11): 1825-1843.

[3] L.J. Erik, T.R. Bohez, “Speech recognition using fractals”, Pattern Recognition, 2001, 34(11): 2227-2243.

[4] P. Zhang, H. Barad, “Fractal dimension estimation of fractional Brownian motion”, IEEE Southeastcon 90 Proceedings, 1990, Vol.3: 934-939.

[5] Y. Dong, “New Way for Speech Segmentation Based on Fractal Theory”, Journal of Shanghai Jiaotong University, 1998, 32(4): 97-99.

[6] X.X. Shi, “Application of a time regulation algorithm in artificial neural network speech recognition”, Journal of Southeast University, 1999, 25(9): 47-51.

Feature Extraction of Bridges over Water in High Resolution Satellite Imagery

Aijun Chen
Department of Information and Communication Engineering, Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: jschenaj@163.com

And

Jinzong Li
Department of Information and Communication Engineering, Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: jinzong@hit.edu.cn

And

Bing Zhu
Department of Information and Communication Engineering, Harbin Institute of Technology
Harbin, Heilongjiang 150001, China
Email: zhubing@hit.edu.cn

ABSTRACT

Feature extraction plays a critical role in pattern recognition. With respect to a bridge over water in high resolution satellite imagery, the most distinct feature is a line pair. In this paper, a method for extracting line pairs of bridges over water is presented. Firstly, denoising is implemented by mean filtering to an original image and a histogram-based threshold is selected to segment the smoothed image. Then two mathematical morphology operators are used to filer the binary image, and the river direction is obtained from the centroids of two river areas in the binary image. At the same time, an edge image is obtained in which two parallel lines are detected which are most perpendicular to the river. Finally, the extracted features are verified by prior knowledge. Experimental results obtained from real high resolution satellite images show that the proposed method is effective.

Keywords: Satellite Imagery, Bridge, Feature Extraction, Parallel Pair, Mathematical Morphology, Gray Histogram.

1. INTRODUCTION

With the development of modern remote sensing technology, the resolution of satellite images is higher and higher, and the application of satellite images is more important. How to process and recognize satellite images becomes an issue in information engineering area.

At present, targets in satellite images are often classified into two classes: those belong to block target, such as aircraft carrier, tank, large block building, etc and those belonging to linear target, such as bridge, road, river, etc [1]. Recently, recognition to linear targets, especially to bridges, attracts more attention in ATR (Automatic Target Recognition) research. And bridge recognition is one typical application. Image recognition is difficult and critical in image processing and vision and the key step is feature extraction. In a sense, feature extraction plays a critical role in recognition results and it affects the following detection and recognition algorithm. Therefore, how to quickly extract effective features in satellite imagery with high resolution is an important issue in satellite image processing.

A method to fast extract features is presented in terms of characteristics of bridges over water in satellite imagery with high resolution, and experimental results derived from real satellite imagery are illustrated in the end.

2. FEATURE EXTRACTION

The properties of bridges over water can be described as follows: (1) The gray values of bridge areas are similar to those of areas outside rivers; (2) A bridge over water crosses over a river; (3) A bridge over water divides a river into two uniform parts. According to these properties, the process of extracting bridge features consists of the following 6 steps.

2.1 Mean Filtering

In the practical application, the data from high resolution satellite imagery are always bothered by some kinds of noise among which the shot noise is dominant [2]. Therefore, the pre-processing, i.e., local filtering operation of the image is necessary before following processing. Owing to its weak correlation, the shot noise can be easily eliminated with many methods, such as mean filtering, statistical smoothing filtering. In this paper, a method called 8-neighborhood mean filtering is adopted.

Let f be the original image and $f(x, y)$ the gray level of pixel (x, y) in the image, then the 8-neighborhood mean of pixel (x, y) can be expressed as

$$m = \frac{1}{8}\left[\sum_{i=-3}^{3}\sum_{j=-3}^{3} f(x+i, y+j) - f(x, y)\right] \quad (1)$$

The processed image g after mean filtering can be defined as

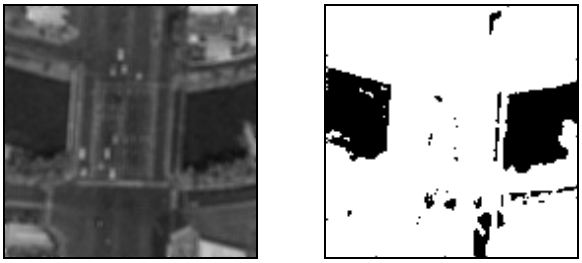
$$g(x, y) = \begin{cases} m & |m - f(x, y)| > T_0 \\ f(x, y) & otherwise \end{cases} \quad (2)$$

where T_0 is a given non-negative threshold.

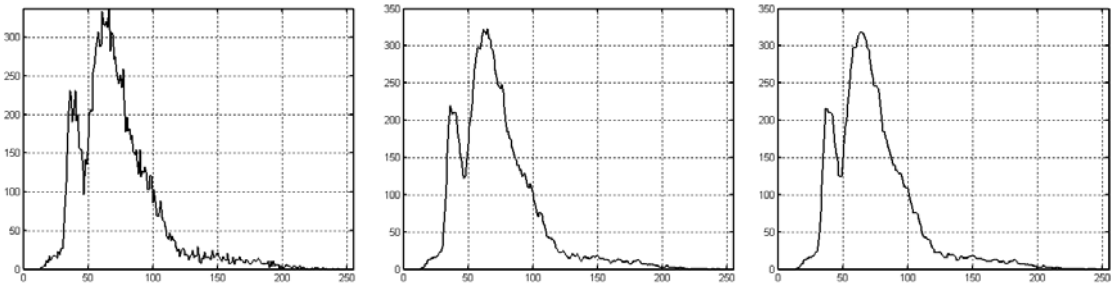
2.2 Threshold Segmentation

Image segmentation is important for it plays a critical role in extracting target feature information. Thresholding has been

a popular tool used in image segmentation [3]. It is useful in separating objects from background, or discriminating objects from objects that have distinct gay-levels. The basic purpose of gray-level thresholding is to extract the 'object' from a given image. One may view it as a classification problem and the image pixels can be classified into two classes: those belonging to the 'object' class and those belonging to the 'background' class. A thresholded image *bw*1 can be created by defining



(a) original image (e) binary image



(b) gay level histogram of original image (c) mean filtering to histogram of (c) (d) median filtering to histogram of (c)

Fig. 1. Image thresholding

$$bw1(x,y)=\begin{cases}1 & f(x,y)>T\\ 0 & f(x,y)\le T\end{cases} \tag{3}$$

where f denotes a original image and T is the global threshold.

In this paper, an optimal threshold is selected to segment a given image. For an image including bridges, it can be divided into two parts: one belongs to river regions whose gray level is low and another belongs to non-river regions whose gray level is high. So there exist two distinct peaks in the histogram of an image with bridges: one with high gray level corresponding to non-river areas and another with low gray level corresponding to river areas. According to the histogram, we can select the valley existing between the two peaks as the threshold to segment the image with bridges. However, there exists local fluctuation in the histogram, which may generate many false peaks and valleys. Therefore, in order to find the proper peaks and valleys, it is necessary to remove the fluctuation before selecting the valley. Here we adopt a smoothing filter and a median filter to smooth the histogram so that the optimal threshold can be properly selected.

Let $h(z)$ be the histogram of an original image, then the smoothed histogram $h_1(z)$ after smoothing filtering can be calculate as

$$h_1(z)=[h(z-1)+h(z)+h(z+1)]/3 \tag{4}$$

and the histogram $h_2(z)$ with median filtering to $h_1(z)$ can be defined as

$$h_2(z)=Median\,[h_1(z-1),\,h_1(z),\,h_1(z+1)] \tag{5}$$

Fig.1. shows a thresholding processing of an image with a bridge. Fig.1. (a) is the original image and Fig.1. (b) is the original histogram of the image in Fig. 1. (a). Figs. 1.(c) and 1(d) illustrate the smoothed histogram with smoothing filtering and the resulting histogram with median filtering, respectively. As seen in Figs. 1.(b), 1.(c) and 1.(d), the histogram is more smoothing and the peaks and the valley are more apparent. The thresholded image by the threshold selected from the valley of the histogram in Fig. 1. (d) is showed in Fig. 1. (e).

2.3 Morphology Filtering

In a thresholded image, it is common that there exist small black blocks in a large white region and small white blocks in a large black region. These small blocks can result in interference in edge detection and line segment extraction. So some measures are adopted to delete those blocks. A method based on mathematical morphology [4] is applied in this paper. Based on the fact that close operation can not only delete some small black blocks, but also eliminate some protuberances and open operation can fill some small white blocks, the binary image *bw*1 is processed by morphology filtering with a 5×5 diamond-shaped structuring element showed in Fig. 2..

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

Fig. 2. The structuring element

The processed image *bw*2 can be obtained by

$$bw2 = bw1 \bullet strel \circ strel \tag{6}$$

where $\bullet$ denotes close operator and $\circ$ is open operator.

2.4 Edge Detection

There are many edge detection methods and a general one is to use an edge detection operator, such as Canny operator, Sobel operator and Prewitt operator, etc, to detect edge in a gray-level image [5]. Owing to the fact that the image is a binary one and it is maybe slow to extract edges from such an image using an edge operator, a fast algorithm [6] is used. The basic idea of this method is to hollow internal pixels of black regions. That is, for all pixels of a black region, if all the gray level of 8 8-neighborhood points of a black point (whose gray level is 0) is 0, set the gray level of the current pixel be 1. It can be expressed by

$$bw3(x,y) = \begin{cases} bw2(x,y) = 0 \ and \\ 0 \quad \sum_{i=-3}^{3}\sum_{j=-3}^{3} bw2(x+i, y+j) \neq 0 \\ 1 \qquad otherwise \end{cases} \tag{7}$$

where *bw*3 represents the edge image and bw2 is the binary image after morphology filtering to a thresholded image.

2.5 Line Segment Detection

Some pixels are labeled as edge points after edge detection and the following work is to extract line segments from the edge points. The most common method to extract segments is based on Hough transform (HT) and its improvement. Thanks to the fact that HT has the disadvantages of memory-consuming and time-consuming, a split-merge method [7] is applied to extract line segments in this paper. The algorithm can be described as follows: The black pixels, namely edge points, are tracked by 8-neighborhood connection relationship between pixels so that a series of edge chains can be obtained. Then each edge chain is processed with the following steps.

Step 1. Initialization. Divide the edge chain into several sects with the same length, L_1, $L_{2...}$ L_n.

Step 2. Splitting. Each sect chain L_i is fitted to line segments by least square method. If the least square residual E_i is larger than a given threshold T_1, divide the current sect chain into two parts at the point whose perpendicular distance to the line segment is the largest. All sect chains are processed according to the above method, then continue.

Step 3. Merging. For two neighboring chains, if the least square residual of the fitted line after merged into one is less than the given threshold T_1, merge them into one.

Step 4. Endpoint adjusting. Inspect endpoints resulted from two neighboring chains. Let v_i be the endpoint obtained from L_{i-1} and L_i. E_{i-1} and E_i denote the least square residuals of L_{i-1} and L_i, respectively. Adjust v_i to v_i' if v_i and v_i' are satisfied with Eq. (8) after v_i is adjusted to v_i' in a small range.

$$\sqrt{E_{i-1}^2 + E_i'^2} < \sqrt{E_{i-1}^2 + E_i^2} \tag{8}$$

Step 5. Stop estimating. During the processing of step 2 to 3, if the edge chain changes, return to step 2 and repeat. Otherwise, stop.

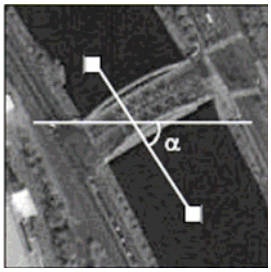


Fig. 3. The direction of a river (α=58°)

2.6 Parallel Pair Detection

The most apparent property of a bridge is that there exist two approximately parallel line segments on both sides of a bridge. Therefore, detecting the parallel pair is a critical step to judge whether there exist bridges on an image or not. Because the angle between the line segment on the side of a bridge and the river is larger, not all parallel pairs are detected, but only those are detected which are approximately perpendicular to the river.

A great deal experimental results show that it is appropriate to regard the angle between horizontal axis and the line passing through centroids of two parts of the river as the direction of the river. As shown in Fig. 3., α is the direction of the river and two white squares represent two centroids of two river regions.

Let β be the angle of a segment. If Eq. (9) is satisfied, the segment is a candidate bridge line segment.

$$\left|\alpha + \beta - 90°\right| < \delta \tag{9}$$

where α is the direction of the river and δ is a positive integer.

Actually the number of candidate line segments at least is two. If the number is just 2, the two candidate line segments are desired. While the number is larger than 2, adjust the angles of candidate line segments and let the mean value of angles be that of each line segment. Two line segments, which are farthest each other among them, are considered as desired ones.

A line with slope-intercept form can be written as

$$y = mx + b \tag{10}$$

where *m* is the slope of the line and *b* is the y-intercept.

Let L_1, L_2… L_n be detected candidate line segments and b_1, b_2, …, b_n be their y-intercepts, respectively. b_i and b_j denote y-intercepts of L_i and L_j, respectively. If Eq. (11) is satisfied, L_i and L_j are desired line segments.

$$\max(|b_i - b_j|) \ (i \neq j; i = 1,2,\cdots,n; j = 1,2,\cdots,n) \tag{11}$$

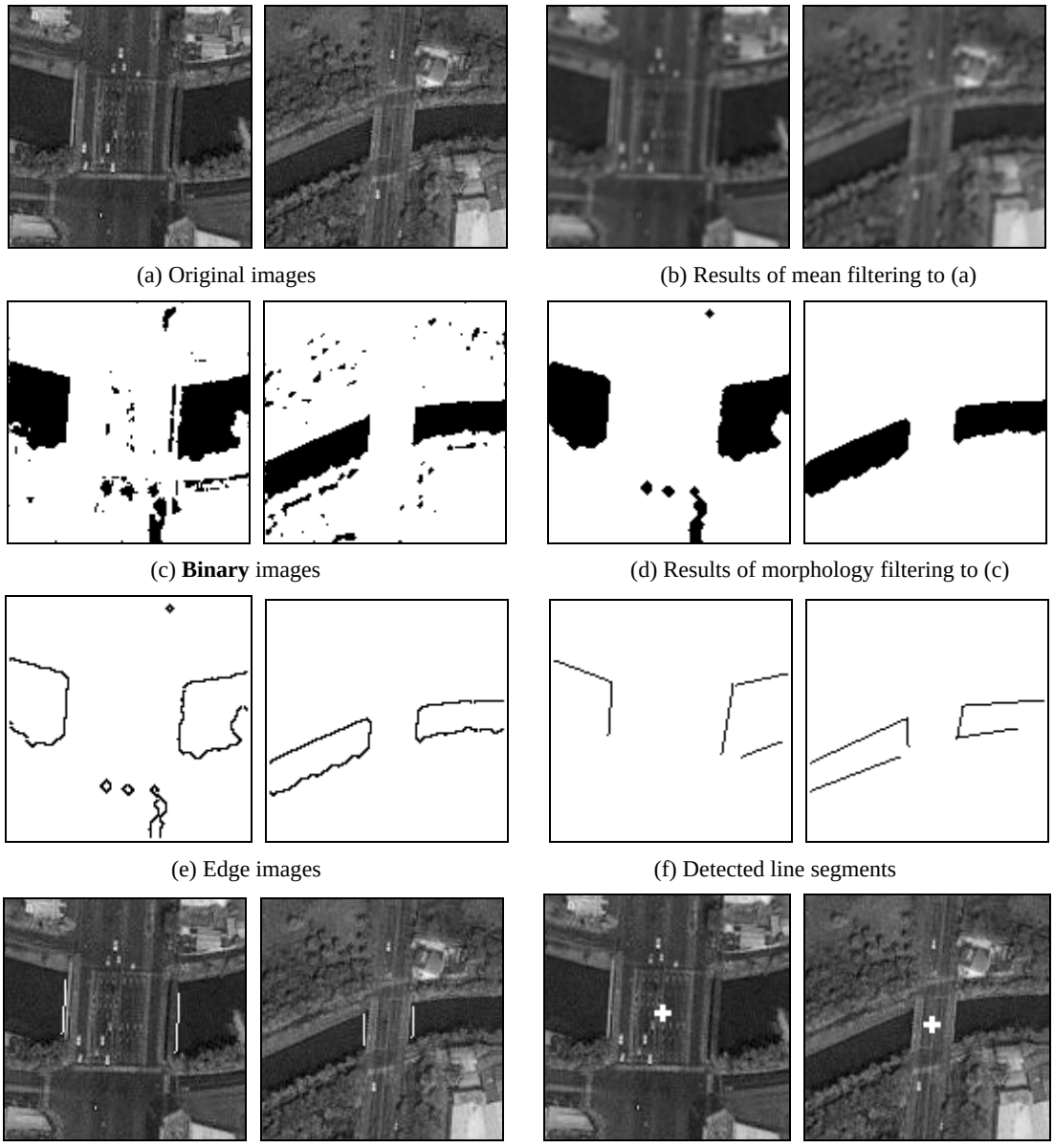
Whether or not the detected pair of line segments is from a real bridge is verified by the following work. Several points are selected from the central section of the area between the two line segments and one line passing through such a point is drawn which is perpendicular to the river. If the points belonging to one line and locating outside the two line segments stand in river areas, the detected pair of line segments can be regarded as bridge lines.

In the end, calculate the midpoint of the line segment whose endpoints are midpoints of the two bridge line segments. The bridge is located at the point.

3. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented in a Pentium 1.60GHz computer with Windows XP operating system using Visual C++ 6.0 program language. In our experiments, 30 real images with size 256×256 and gray level 256 have been tested. The mean execution time is 0.03s.

Two typical experimental results are shown in Fig. 4.. The original images are shown in Fig. 4. (a), and each of them includes a bridge. In Figs. 4.(b)-(g), a series of processing results are shown. Fig. 4.(h) shows the locations of bridges in the images.



(a) Original images (b) Results of mean filtering to (a)

(c) **Binary** images (d) Results of morphology filtering to (c)

(e) Edge images (f) Detected line segments

(g) Results of parallel pair detection (h) Bridge images with location signs

Fig. 4. Typically experimental results

4. CONCLUSIONS

A simple and efficient approach has been presented for extracting features of bridges over water in high resolution satellite imagery. It consists of the following six steps: mean filtering, thresholding, morphology filtering, edge detection, parallel pair detection and feature verification. In this paper, we make full use of prior knowledge to extract features. With the knowledge that the gay level of river areas is different from that of non-river areas and that there are two apparent peaks in the gray level histogram, the valley between the two peaks is selected as the threshold to segment an image. What's more, the fact that the angle between a river and a bridge over it is larger is applied to detect candidate bridge lines. That is, not all parallel pairs are detected, but those are selected which are appropriately perpendicular to the river. Experimental results show that the proposed method is fast, efficient and can be used in real

world.

If a river is dry or a bridge is not over a river, the proposed method will not work well for it is made to extract features of bridges only over water. How to improve the proposed method to be widely used is the future research issue.

5. REFERENCES

[1] H.Wu, Z.K.Liu, et al, “A Study of Bridge Recognition from Landsat TM Images”, Journal of Remote Sensing, 2003, 7(6):478-484.

[2] X.H.Yuan, L.Z.Jin, et al, “Recognition of Bridges over Water through Detecting and Analyzing Regions of Interest”, Journal of Infrared and Millimeter Waves, 2003, 22(5):331-336.

[3] Y.J.Zhang, Image Segmentation, Beijing: Science Press, 2001.

[4] E.J.Breen, R.Jones, et al, “Mathematical Morphology: A Useful Set of Tools for Image Analysis”, Statistics and Computing, 2000, 10(2):105-120.

[5] Y.D.Qu, C.S.Cui, et al, “A Fast Subpixel Edge Detection Method Using Sobel–Zernike Moments Operator”, Image and Vision Computing, 2005, 23(1):11-17.

[6] B.He, T.Y.Ma, et al, Digital Image Processing with Visual C++, Beijing: Posts and Telecom Press, 2001.

[7] N.N.Zheng, Computer Vision and Pattern Recognition, Beijing: National Defense Industry Press, 1998.

A Multiscale Edge Extracting Technique Based on the Fuzzy Rules *

Ruihua Lu
School of Electronics and Information Engineering, Southwest University,
Chongqing, 400715, China
Email: chenlr@swu.edu.cn

And

Ming Yang
School of Computer and Information Science, Southwest University,
Chongqing, 400715, China
Email: yangming@swu.edu.cn

ABSTRACT

In this paper a multiscale edge extracting technique based on the fuzzy rules is proposed. The cardinal thought of the technique is that edges can be gotten usually via extracting large-amplitude parts of signal changes in neighboring pixels. However, it is very difficult to get a threshold for distinguishing the large signal differences from small ones, because image characteristics are ambiguous and the threshold should be varied according to some local characteristics of the image. The proposed technique can perform a proper control of the threshold for separating the small-scale output relying on the value of the large-scale output. Here, the fuzzy rules are used to consider the unstableness and uncertainty of the signals. Multiscale edge extracting can obtain multiscale edges, which contain more detail information in comparison with monoscale edges. Besides, edge tracing can be performed to obtain more precise edges by dividing the amplitude of the output into three components with two thresholds and making a decision with the fuzzy rules. Computer simulations verify the effective performance of the proposed edge extracting technique.

Keywords: Multiscale edge extracting, Fuzzy rule, Threshold control, Signal difference, Differential operator.

1. INTRODUCTION

Usually, an image is a representative unstable signal. In processing such an image unstableness must be taken into account. There are some effecttive methods to gain stability. One of them is to control the system parameters relying on the local characteristics of the image [1, 2]. In using the method the system must discriminate flat regional parts of the image from steeply varying ones such as edges, and then set the system parameters for each part separately. But it is difficult for such control to be performed specifically, because the definition of the signal characteristics and the rules describing how to control the system parameters are often represented in an uncertain form. For instance, in some part of the edge the signal difference between two neighboring pixels is small while in some part of the flat region the signal difference is large because of noises. But it is very hard to get a definite threshold for discrimination of small differences from large ones. Besides, the window size of the noise reduction filter must be large enough in the flat region, while it must be small enough in the part of edges. But it is rather difficult to define the large and small window size exactly. Therefore, ambiguity must be taken into consideration in edge extraction. In this paper a multiscale edge extracting technique based on the fuzzy rules is proposed. The paper consists of 5 sections in addition to introduction. Section 2 presents cardinal thought of edge extraction based on the fuzzy rules. Section 3 introduces fuzzy rule-based edge extraction using multiscale edge images. Section 4 shows how to get a binary edge image. Section 5 demonstrates computer simulations. Section 6 makes a conclusion.

* Supported by the fund of University Science and Technology under Grant SWNU 2004006.

2. CARDINAL THOUGHT OF EDGE EXTRACTION USING THE FUZZY RULES

Edges can be gotten usually via extracting large-amplitude parts of signal changes in neighboring pixels. When the signal change is large at the pixel (i,j), an edge is assumed to exist there, but when the signal change is small at the pixel (i,j), an edge does not exist there. Here, how to distinguish large signal differences from small ones is still an open question. The threshold for distinguishing them is difficult to be gotten, because image characteristics are ambiguous. Besides, the threshold should be varied according to some local characteristics of the image. For example, if there is a weak edge in a rough area, the threshold should be large, while if there is a weak edge in a smooth area, the threshold should be small. So a system based on the fuzzy rules should be applied. Assume that the value of the signal change at the pixel (i,j) is expressed as $u(i,j)$, the output of the edge extracting system $y(i,j)$ can be represented as:

$$y(i,j) = F[u(i,j), X(i,j)], \quad (1)$$

where F is a binary function with a value either 0 or 1 ($F[\cdot,\cdot]=1$ means existence of an edge at pixel (i,j); $F[\cdot,\cdot]=0$ means nonexistence of an edge at the pixel); $X(i,j)$ is a set of input image signals about the pixel (i,j), and $y(i,j)$ is obtained from the values of $u(i,j)$ and $X(i,j)$. If edges can be specifically decided from the value u (i,j), $y(i,j)$ can be a function of only $u(i,j)$ as shown in Fig. 1.. However, as image signals are unstable, the threshold in Fig. 1. varies relying on other local characteristics in $X(i,j)$. Here a parameter $\omega(i,j)$ is introduced indicating to what extent the pixel (i,j) is accounted to be an edge point.

The $\omega(i,j)$ has a continuous value from 0 to 1; The pixel is an edge point when the value is large. By cutting the value ω (i,j) with a threshold, binary edge images are gotten. As will be seen in Section 3, the parameter $\omega(i,j)$ is obtained from fuzzy approximate reasoning from various local characteristics

about the pixel (i,j) including $u(i,j)$, and, correspondingly, the $\omega(i,j)$ is represented as a nonlinear function of these local characteristics.

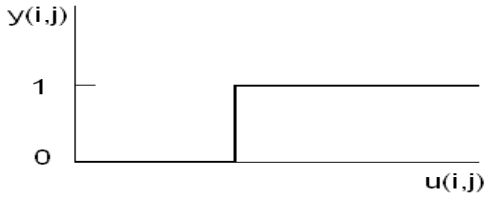


Fig. 1. The signal value $y(i,j)$ of the edge image as a function of the signal change $u(i,j)$ when the edges are explicit.

3. MULTISCALE EDGE EXTRACTION BASED ON THE FUZZY RULES

In recent years multiscale analysis has attracted attention of the scientists considering it as a way to solve the problem of nonstationarity. According to their opinion, in this analysis multiple differential operators with different window sizes process an image in order to get multiscale edges. If the window size is large, rough edges are gotten, which indicate large structures of the image objects but do not indicate fine structures. If the window size is small, fine edges containing both the fine structures and noisy components are gotten. Combining these multiscale edges allows us to make a more exact judgement on the edges than that by using monoscale edges. Consequently, such multiscale edge images, in other words, the outputs of multiscale differential operators can be the local characteristics which are efficient to decide the value $\omega(i,j)$for exact edge extraction [3].

The value $\omega(i,j)$ can be gotten from the multiscale edge images via the following fuzzy rule:

Rule 1: if small-scale edge is weak and large-scale edge is weak, then $\omega(i,j)$ is small;

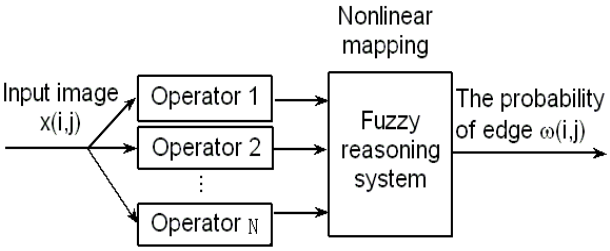
If small-scale edge is strong and large-scale edge is strong, then $\omega(i,j)$ is large;

else $\omega(i,j)$ is medium.

Here, the parameter $\omega(i,j)$ is represented as a continuous nonlinear function of the small and large scale edge images. The signal difference u (i,j) accords to the small-scale edge image here. Because it is unknown how small or large the window size should be for the small-scale and the large-scale edge images, a filter bank of several differential operators with different window sizes such as 3×3, 5×5, 7×7, is readied. The block diagram of this system is given in Fig. 2.

The function $\omega(i,j)$ is approached by a optimized step-like function [4], thus the mean square error between the system output $\omega(i,j)$ and the hoped edge image $h(i,j)$ for the image to be processed $x(i,j)$ will be the minimum. Here, the hoped edge image is composed of binary signals 0 and 1; if the pixel (i,j) is an edge point, $h(i,j)$ has a value 1, but if the pixel is not an edge point, $h(i,j)$ has a value 0.

Presume that $\omega(i,j)$ is expressed as ω_{pq} when $o_1(i,j)$, i.e. the output of Operator 1, is in the p-th divided region and $o_2(i,j)$, i.e. the output of Operator 2, is in the q-th divided region. The system output $\omega(i,j)$ has the following equation:



$$\omega(i,j)=\sum_{p=1}^{P}\sum_{q=1}^{Q}C_{ijpq}\,\omega_{pq}, \qquad (2)$$

Fig. 2. Block diagram of the fuzzy edge extracting system by multiscale edge images.

where

$$C_{ijpq}=\begin{cases}1 & \text{if } o_1(i,j) \text{ is in the } p\text{-th divided region}\\ & \text{and } o_2(i,j) \text{ is in the } q\text{-th divided region.}\\ 0 & \text{else}\end{cases}$$

The mean square error between ω (i,j) and $h(i,j)$ is represented as:

$$\begin{aligned}E &= E[(\omega(i,j)-h(i,j))^2]\\ &= E[(\sum_{p}\sum_{q}C_{ijpq}\,\omega_{pq}-h(i,j))^2].\end{aligned} \qquad (3)$$

Finally, ω_{pq} minimizing E is gotten from $\frac{\partial E}{\partial \omega_{pq}}=0$ and expressed as follows:

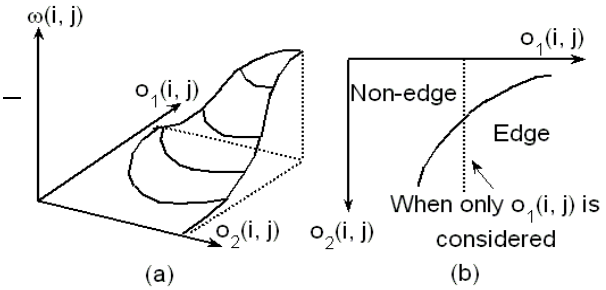
$$\omega_{pq} = E_{pq}[h(i,j)], \qquad (4)$$

where $E_{pq}[h(i,j)]$ indicates the average of $h(i,j)$ only under the circumstance that $o_1(i,j)$ and $o_2(i,j)$ is in the p-th and q-th divided region, respectively. This interpretation is only made for the Operator 1 and 2, but this approach can be easily expanded to more dimensions with more differential operators.

4. GETTING A BINARY EDGE IMAGE

The parameter $\omega(i,j)$ has continuous values from 0 to 1. The certain binary edge image $y(i,j)$ can be gotten by lessening the ω (i,j) with a threshold. For example, when ω (i,j) is represented as in Fig. 3. (a), the final edge image can be gotten via separating the feature space nonlinearly as demonstrated in Fig. 3.(b), where the left side of the solid curve is inferred as the non-edge part. For reference, when only the output of the small-scale differential operator o_1 (i,j) is considered, the separation of the space is shown by a dotted straight line whose left side is inferred as the non-edge part and the right side is as the edge part. It can be seen that the multiscale system based on the fuzzy rules can perform a proper control of the threshold for separating the small-scale output $o_1(i,j)$ relying on the value of the large-scale output $o_2(i,j)$, for the value of $o_1(i,j)$, making inference about edginess, is varied relying on the value of $o_2(i,j)$. Besides, as the edges usually connect to each other, edge tracing can be performed to obtain more precise edges.

First, divide the amplitude of the output into three components by two thresholds T_1 and T_2 ($T_1 > T_2$). Then make



a decision with the following rule:

Fig. 3. Getting a binary edge image: (a) an example of the value $\omega(i,j)$as a function of two feature parameters $o_1(i,j)$ and $o_2(i,j)$; (b) nonlinear separation of the feature space

Rule 2: if $\omega(i,j) > T_1$

then decide the pixel (i,j) as an edge point;

if $\omega(i,j) \leqslant T_2$

then decide the pixel (i,j) as a non-edge point;

if $T_1 \geqslant \omega(i,j) > T_2$ and the pixel (i,j) is connected to an edge point

then decide the pixel (i,j) as an edge point.

5. COMPUTER SIMULATIONS

Computer simulations are realized by Robinson's operator [5] as the differential operator, because it takes the direction of the edge into account. The Robinson's operator affords the gradient in four directions (a horizontal one, a vertical one and two sloping ones in 45° *and* 135°) by the following eight 3×3 matrices [6]:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

Each of these matrices is multiplied to the 3×3 input image round each pixel (i,j), i.e. if

$$G = \begin{bmatrix} g_{-1,-1} & g_{-1,0} & g_{-1,1} \\ g_{0,-1} & g_{0,0} & g_{0,1} \\ g_{1,-1} & g_{1,0} & g_{1,1} \end{bmatrix}$$

indicates such a matrix, we have

$$Z_{ij} = \sum_{k=-1}^{1} \sum_{m=-1}^{1} g_{k,m} \cdot x(i-k, j-m). \qquad (5)$$

The maximum amplitude, i.e. the maximum absolute value, in these Z_{ij}'s for the 8 matrices is considered as the output of the operator. Here, as the amplitude is taken, only the upper 4 matrices are used.

The filter bank includes three differential operators. The first one is sized 3×3 and accords to the usual Robinson's operator. The second is sized 7×7, corresponding to an expansion of the Robinson's operator to a 7×7 window.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -2 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & 2 & 1 & 0 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -2 & -1 & -1 \\ 1 & 0 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 2 & 0 & -2 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 1 & 2 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -2 & -1 & 0 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 0 \end{bmatrix}$$

The third differential operator is a large one. As the window larger than 7×7 engrosses a lot of computation time, a one-dimensional derivative operator sized 1×11 is applied as the third differential operator for simplicity. The output of the third one has the following expression:

$$o_3(i, j) = \left| \sum_{k=1}^{5} z(k) - \sum_{k=1}^{5} z(-k) \right|. \qquad (6)$$

Here, $z(k)$ and $z(-k)$ denote the values of the input image $x(i,j)$ which are aligned perpendicular to the direction of the edge detected by the 3×3 operator as seen in Fig. 4.. Under this circumstance the number of the pixels accounted in smoothing is lessened, but wider regions perpendicular to the edge can be taken into account.

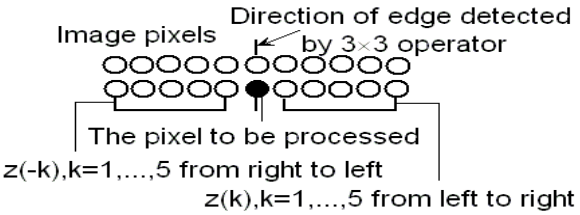
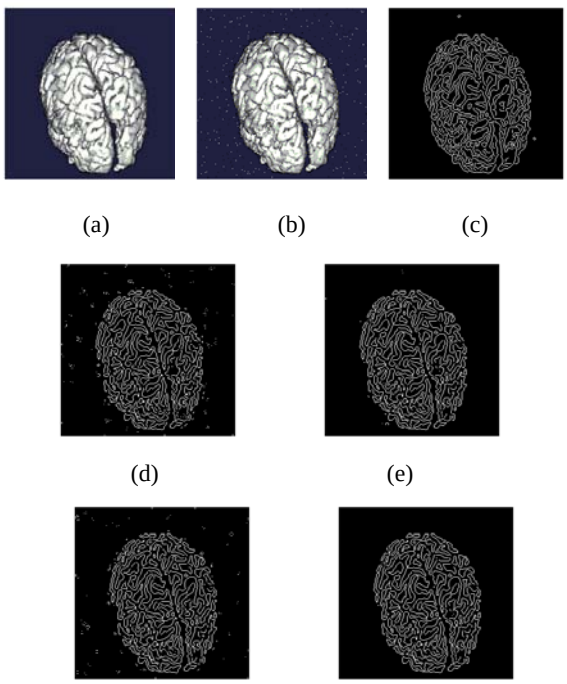


Fig. 4. Pixels used in the one-dimensional operator. The shaded pixel is used as $z(k)$ and $z(-k)$.

Computer simulations of the edge detecting system for a test image is shown in Fig. 5.: (a) is the input image, (b) is the image corrupted by noises, (c) is the hoped edge image which is made by processing an image like (b) with the 3×3 Robinson's operator and by selecting the pixels where the output surpasses a threshold as the edge part, (d), (e) and (f) are the outcomes of edge extraction by differential operators with window sized 3×3, window sized 7×7, and the one-dimensional window sized 1×11, correspondingly. It can be seen that when the window size is small, the noisy components are extracted as edges too, but when the window size is large, the edges round fine structures such as thin lines are omitted, (g) is the output of the proposed in this paper fuzzy rule-based system for multiscale edges.



(f) (g)

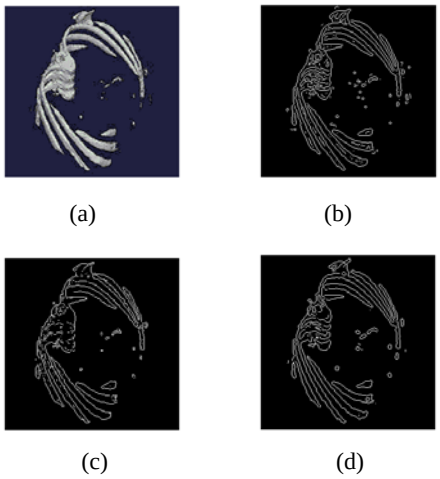
Fig. 5. The outcomes of edge extraction for a test image by differential operators

It is obvious that the edges round fine structure can be extracted, while the noisy components are rejected. Here two-level thresholding is applied in order to obtain the binary edge image.

Fig. 6. shows the output for a natural image: (a) is the input image, (b) and (c) are the outcomes of edge extraction by a single derivative operator, sized 3×3 and 7×7, correspondingly, (d) is the outcome of the system proposed in this paper. Edge tracing is realized for getting the binary edge image in (b), (c) and (d). It can be seen that the background noises are much more rejected in (d) while the edge lines are kept clear.

6. CONCLUSION

A multiscale edge extracting technique based on the fuzzy rules is proposed. Cardinal thought of edge extraction using the fuzzy rules consists in getting edges via extracting large amplitude parts of signal changes in neighboring pixels. In order to get the threshold for distinguishing large signal differences from small ones a system based on the fuzzy rules should be applied. In such a system multiscale edge images for edge extraction are used. The multiscale edge images, i.e. the outputs of multiscale differential operators, can be the local characteristics, which are efficient to decide the value ω (i, j) for exact edge extraction. By cutting the value $\omega(i,j)$ with a threshold binary edge images are obtained. Computer simulations performed by Robinson's operator as the differential operator show that the proposed edge extracting system based on the fuzzy rules affords a powerful method for edge extracting in image processing.



(a) (b)

(c) (d)

Fig. 6. The output for a natural image

7. REFERENCES

[1] S.A. Rajala, R.J. DeFigueiredo,"Adaptive nonlinear image restoration by a modified Kalman filtering approach," IEEE Trans. Acoustic Speech Signal Processing, vol.29, no.5, 1981, pp.1933-1041.

[2] C.A. Pomalaza-Raez, C.D. McGillem," An adaptive, nonlinear edge-preserving filter," IEEE Trans.Acoustic Speech Signal Processing, vol 32, no.3, 1984,pp.571-576.

[3] K. Arakawa," Fuzzy rule-based edge detection using multiscale edge images," in: Proceedings of the 6th IEEE International Workshop on Intelligent Signal Processing and Communication System, 1998, pp.204-208.

[4] D.Lisin, M.A.Gennert,"Optimal function approximation using fuzzy rules," in: Proceedings of IEEE International Conference North American Fuzzy Information Processing Society, New York, Jan.10-12, 1999, pp.184-188.

[5] G.S. Robinson, "Edge detection by compass gradient masks," Computer Graphics Image Processing, 1977, 6 (5), pp.492-501.

[6] Santiago Aja-Fernandez, Carlos Alberola-Lopez,"Fast Inference Using Transition Matrices: An Extension to Nonlinear Operators," IEEE Trans. Fuzzy Systems, vol.13, no.4, 2005, pp.478-490.

Image Compression Based on the Improved SA Algorithm

Yiqian Tang
Chengdu University,Chengdu, 610106, China
Email:yqt111@tom.com
Chunli Feng
Liaoning Institute of Technology, Jinzhou, 121001, China
Email: fchunli1981@163.com
Yue Zhao, Lin Shi
Chengdu University,Chengdu, 610106, China
Email:lgzhytty@tom.com

ABSTRACT

Aiming at the techniques of the image compression, we put forward a kind of improved local learning adaptive Silva-Almeida (SA) algorithm on the basis of BP neural networks. The training process of the new algorithm was divided into the acceleration and steady convergence periods, and we adjusted the learning rate and the momentum factor locally at the same time. By the experiment of the curve fitting, the improved SA algorithm quickened the training speed, raised precision distinctly, and resisted the oscillation in a large scale. In the test of the image compression, compared with the resilient propagation (RPROP) method, the new algorithm's training speed and the quality of the reconstruction images were enhanced very greatly, and we acquired a higher ratio of image compression. Therefore the improved SA algorithm is more effective to the image compression.

Keywords: SA algorithm, image compression, BP networks, resilient propagation, neural networks.

1. INTRODUCTION

D. E. Rumelhart, G. E. Hinton and R. J. Williams put forward the error back-propagation (BP) algorithm in 1986[1]. After that, the BP networks became one of the most extensive artificial neural networks in application currently. It has many merits, such as solid theoretical foundation, deducing process strictly and applying generally. These make it have special advantages in the aspect of image compression [2]. But the standard BP algorithm put forward by Rumelhart exists some problems. Its convergence speed is lower. It is easy to be trapped into local minimum value and sensitive to the initialized value. Because of these weaknesses in the BP algorithm and the other improved algorithms at present, there are a lot of places to be solved in the application of the image compression all the same.

The method of learning rate adjustment is a very valid improved method of the BP algorithm. It includes the Local Learning Rate Adaptation method and the Global Learning Rate Adaptation method [3]. The Local Learning Rate Adaptation method has a lot of ways. For example, Silva and Almeida provided the method that adjusted its learning rate according to the symbol of the gradient of the connecting weight values to the objective function. It is called SA method. It is equal to RPROP method, which was provided by Riedmiller in convergence accuracy, but its convergence speed cannot compare with RPROP method.

Nowadays, the intelligent theory is applied widely. As an intelligent algorithm, an artificial neural network has an important effect. In order to satisfy the image compression, we proposed a new kind of learning method, which had short training time, and high quality of reconstruction image. The improved local adaptive SA algorithm resists the oscillation in a large scale and accelerates the pace of training. The simulation tests have proven that the quality of the reconstruction images are improved markedly.

2. SA ALGORITHMS AND IMPROVED SA ALGORITHM

2.1 SA algorithm

The adjustment of learning rates should be based on gradients of the learning rate according to the objective function. Silva's method adjusts the learning rates based on the symbols of the current and the previous gradient of connecting weights of the objective function. If the symbols change, the corresponding learning rates will reduce. If the symbols don't change, the corresponding rates will increase. The algorithm is described as follows [4]:

(i) For each corresponding connecting weights to learning rate initialize a small random values: $\eta_{ij}(0)$.

(ii) Adjustment of learning rates:

$$If \ \frac{\partial E}{\partial \omega_{ij}}(n) * \frac{\partial E}{\partial \omega_{ij}}(n-1) \geq 0, \eta_{ij}(n) = \eta_{ij}(n-1) * u \quad (1)$$

$$If \ \frac{\partial E}{\partial \omega_{ij}}(n) * \frac{\partial E}{\partial \omega_{ij}}(n-1) < 0, \eta_{ij}(n) = \eta_{ij}(n-1) * d \quad (2)$$

(iii) Adjustment of connecting weights

$$\Delta\omega_{ij}(n) = -\eta_{ij}(n) * \frac{\partial E}{\partial \omega_{ij}}(n) \quad (3)$$

Methods for the establishment of parameters: $u \approx \frac{1}{d}$, $0 < d < 1 < u$, scopes of u and d are 1.1~1.3 and 0.7~0.9.

2.2 Improved SA algorithm

In the SA algorithm, the larger value of u is, the faster learning rate will increase. And the adjustment step of the connecting weights will increase too. In reverse, the learning rates' increase will slow down. But, if we choose a larger value of u, it is easy to make the learning process take on an oscillation phenomenon, and even a decentralization phenomenon in some cases. A smaller value of u is chosen in a general way to ensure the stability of the learning process. When the initial learning rate ($\eta_{ij}(0)$) is a smaller

value, it is inevitable result in the slow speed of learning.

In order to improve the algorithm, the learning process can be divided into two stages. The first phase is the acceleration period. At this time, a larger value of the *u* is chosen to let the learning rate increase around the best value quickly. Then the learning rates are adjusted through a second period of the steady convergence period. The algorithm can restrain oscillation phenomenon in a way, and it also can make learning speed faster than the original SA algorithm. But it is still very sensitive to the initialized weight-values, some still oscillates, and is easy to be decentralization.

Now, we can consider the global learning adaptation method of increasing momentum in the weight-values.

$$\Delta\ \omega_{ij}(n) = -\eta\frac{\partial E}{\partial \omega_{ij}} + \alpha\,\Delta\omega_{ij}(n-1) = -\eta\sum_{t=0}^{n}\alpha^{n-1}\frac{\partial E(t)}{\partial \omega_{ij}(t)} \quad (4)$$

From the above formula, we can see that when the current and the previous $\frac{\partial E(t)}{\partial \omega_{ij}(t)}$ have the same symbols, the sum of weight-values increases. So the $\Delta\omega_{ij}(n)$ is larger, and we can get a steady-going adjustment and increase the speed of adjustment for ω. When the current and the previous $\frac{\partial E(t)}{\partial \omega_{ij}(t)}$ have the contrary symbols, that is to say, there is a certain oscillation. The sum of weight-values decreases, and it makes $\Delta\omega_{ij}(n)$ reduced. So the method plays a stabilizing role at this time.

Therefore, in order to restrain the oscillation phenomenon further, we improve the SA algorithm by integrating the above two characteristics. When the symbols of gradient of the weight-value according to two consecutive objective functions are same, we examine the ratio of the current gradient value and the previous gradient value. If this ratio is greater than a certain threshold, the adjustment step of weight values is that adjustment step one moment before. The learning rates decrease and momentum factors increase.

In summary, the new improved SA algorithm is described as follows:

(i) Stochastic initial small learning rates: $\eta_{ij}(0)$.

(ii)Local learning adaptation and renewal weight–values .

First stage: Acceleration period

If $\frac{\partial E}{\partial \omega_{ij}}(n) * \frac{\partial E}{\partial \omega_{ij}}(n-1) \ge 0$

If $\frac{\frac{\partial E}{\partial \omega_{ij}}(n)}{\frac{\partial E}{\partial \omega_{ij}}(n-1)} \ge T_1$

$$\Delta\omega_{ij}(n) = \Delta\omega_{ij}(n-1) + \alpha_{ij}(n) * \Delta\omega_{ij}(n-1) \quad (5)$$

$$\eta_{ij}(n) = \eta_{ij}(n-1) * d_3 \quad (6)$$

$$\alpha_{ij}(n) = \alpha_{ij}(n-1) * d_3 \quad (7)$$

Else if $\frac{\frac{\partial E}{\partial \omega_{ij}}(n)}{\frac{\partial E}{\partial \omega_{ij}}(n-1)} < T_1$

$$\eta_{ij}(n) = \eta_{ij}(n-1) * u_1 \quad (8)$$

$$\alpha_{ij}(n) = \alpha_{ij}(n-1) * d_1 \quad (9)$$

$$\Delta\omega_{ij}(n) = -\eta_{ij}(n) * \frac{\partial E}{\partial \omega_{ij}}(n) + \alpha_{ij}(n) * \Delta\omega_{ij}(n-1) \quad (10)$$

Else if $\frac{\partial E}{\partial \omega_{ij}}(n) * \frac{\partial E}{\partial \omega_{ij}}(n-1) < 0$

$$\eta_{ij}(n) = \eta_{ij}(n-1) * d_1 \quad (11)$$

$$\alpha_{ij}(n) = \alpha_{ij}(n-1) * u_1 \quad (12)$$

$$\Delta\omega_{ij}(n) = -\eta_{ij}(n) * \frac{\partial E}{\partial \omega_{ij}}(n) + \alpha_{ij}(n) * \Delta\omega_{ij}(n-1) \quad (13)$$

Now turn into the second stage.

Second stage: Steady convergence period.

If $\frac{\partial E}{\partial \omega_{ij}}(n) * \frac{\partial E}{\partial \omega_{ij}}(n-1) \ge 0$

If $\frac{\frac{\partial E}{\partial \omega_{ij}}(n)}{\frac{\partial E}{\partial \omega_{ij}}(n-1)} \ge T_2$

$$\Delta\omega_{ij}(n) = \Delta\omega_{ij}(n-1) + \alpha_{ij}(n) * \Delta\omega_{ij}(n-1) \quad (14)$$

$$\eta_{ij}(n) = \eta_{ij}(n-1) * d_4 \quad (15)$$

$$\alpha_{ij}(n) = \alpha_{ij}(n-1) * d_4 \quad (16)$$

Else if $\frac{\frac{\partial E}{\partial \omega_{ij}}(n)}{\frac{\partial E}{\partial \omega_{ij}}(n-1)} < T_2$

$$\eta_{ij}(n) = \eta_{ij}(n-1) * u_2 \quad (17)$$

$$\alpha_{ij}(n) = \alpha_{ij}(n-1) * d_2 \quad (18)$$

$$\Delta\omega_{ij}(n) = -\eta_{ij}(n) * \frac{\partial E}{\partial \omega_{ij}}(n) + \alpha_{ij}(n) * \Delta\omega_{ij}(n-1) \quad (19)$$

Else if $\frac{\partial E}{\partial \omega_{ij}}(n) * \frac{\partial E}{\partial \omega_{ij}}(n-1) < 0$

$$\eta_{ij}(n) = \eta_{ij}(n-1) * d_2 \quad (20)$$

$$\alpha_{ij}(n) = \alpha_{ij}(n-1) * u_2 \quad (21)$$

$$\Delta\omega_{ij}(n) = -\eta_{ij}(n) * \frac{\partial E}{\partial \omega_{ij}}(n) + \alpha_{ij}(n) * \Delta\omega_{ij}(n-1) \quad (22)$$

For parameter u_1, the value should be bigger somewhat and its suggested value is 1.5 ~ 3. $d_1 \approx \frac{1}{u_1}$, its suggested value is 0.3 ~ 0.7. For parameter u_2, the value should be smaller somewhat and its suggested value is 1.1 ~ 1.3. The suggested value of parameter d_2 is 0.5 ~ 0.9. The value scopes of parameter T_1 and T_2 need considering both the convergence speed and oscillation phenomena. Their suggested values are 2.5 ~ 4. The suggested values of

parameter d_3 and d_4 are 0.7 ~ 0.9.

2.3 The curve fitting simulation experiments

We compare the new improved algorithm with the method, which only adjusts learning rates locally in the curve fitting application. The training input sample collection is: $u = 0:0.05*pi:0.95*pi$ .The expected output function is $f(u) = \sin(u)$ shown as Fig. 1..

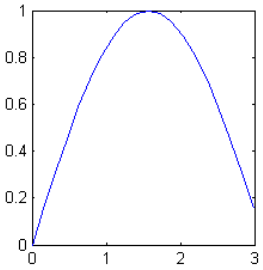
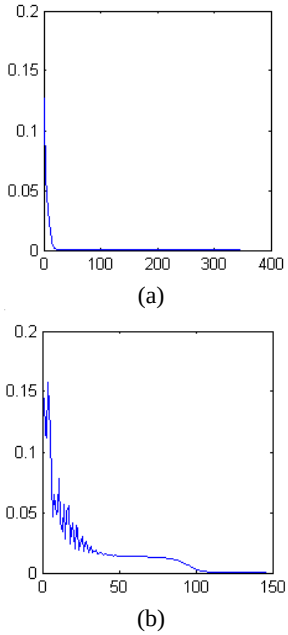


Fig. 1. Expected output function

Now we use three-layers BP network to approach this function. Namely the input layer and the output layer take 1 point. The hidden layer takes 3 points. The function of hidden layer's points and the output point is the Sigmoid function: $f(x) = \frac{1}{1+e^{-x}}$ The parameter's values are set as follows: u_1 =2.5, u_2 =1.2, d_1 =0.4, d_2 =0.6, T_1 =3.0, T_2 =3.0, $d_3 = d_4$ =0.7. The simulation program is realized by the matlab 6.5 procedure. The mean square error is 0.00001. We adopt the training way in groups. The curves of error and the expectation function simulations are shown as Fig. 2. and Fig. 3..

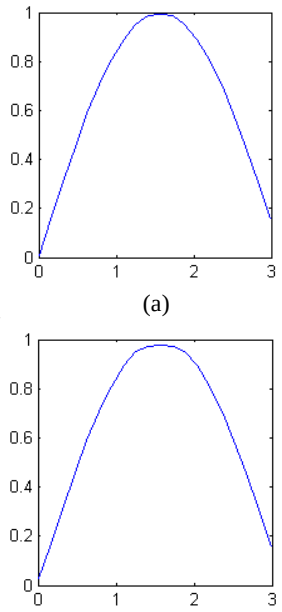


(a)

(b)

Fig. 2. The curves of error
(a) The curve of improved SA algorithm
(b) The curve of method, which only adjusts learning rate locally

Thus the convergence speed of the improved SA algorithm in this article enhanced obviously. It achieved steady convergence in 25 generations basically. In addition, the oscillation phenomena disappear basically in the preliminary training stage. The error function drops smoothly. Further more, we may see from the simulation figure that the training precision higher than the algorithm which only adjusts learning rate locally.



(a)

(b)

Fig. 3. The curves of simulation
(a) Improved SA algorithm in the article
(b) The method which only adjusts learning rate locally

3. IMAGE COMPRESSION BASED ON THE IMPROVED SA ALGORITHM

At present, the neural network is used in image compression widely [5, 6, 7, 8].

The three layer BP (Back-Propagation) network is used mostly, image compression theory [9, 10] is shown as Fig. 4..

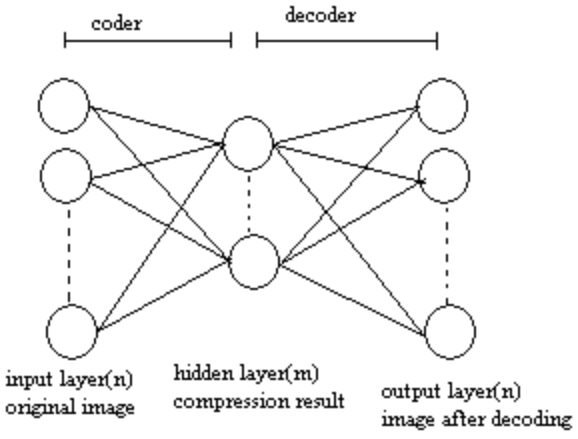


Fig. 4. Image compression theory based on BP network

Adjusting network weight made reconstructed image simulate to original image. From Fig. 4., input layer and hidden layer's weights are equal to a coder, hidden layer and output layer's weights are equal to a decoder. The original image data from input layer are gained in hidden layer through BP network processing; the output data are code compression of original image. Compression ratio of code

image used in BP network is input layer nodes (*n*)/hidden layer nodes (*m*).

In usual situation, in order to speed up the training speed, we often use RPROP algorithm, which the speed of convergence is quick in the image compression. Now, let we compare it with the new SA algorithm.

In the experiment, we use pepper (128×128) to carry on the compression test. Fist we divide pepper image data into the 8×8 the block data, and then establish the network to carry on the training. In the experiment, we have carried on 8 times, 16 times, 32 times and 64 times of image compressions separately and also made the comparison separately with the RPROP algorithm results. The contrast results of their peak value signal to noise ratio (PSNR) and signal to noise ratio (SNR) are shown as table 1.

$$PSNR = 10\lg\frac{255^2}{D} \tag{23}$$

$$SNR = 10\lg\frac{\sigma^2}{D} \tag{24}$$

$$D = \frac{1}{N\times N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\left(x_{mn} - \hat{x}_{mn}\right)^2 \tag{25}$$

$$\sigma^2 = \frac{1}{N\times N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}\left(x_{mn} - \bar{x}\right)^2 \tag{26}$$

$$\bar{x} = \frac{1}{N\times N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x_{mn} \tag{27}$$

In the above formula, x_{mn} is the grey value of pixel (*m, n)* in the original image, and $\hat{x}_{mn}$ is the grey value of pixel (*m, n)* in the reconstruction image.

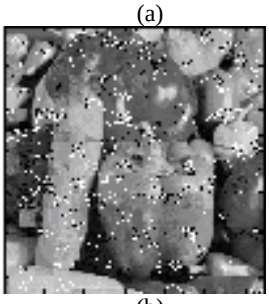
The reconstruction images after compressed by RPROP algorithm are shown as Fig. 6.. The reconstruction images after compressed by the improved SA algorithm are shown as Fig. 7.. Among the reconstruction images compressed by the RPROP algorithm, when the image compression ratio achieved 32 times, the image is not seen clearly. Thus it can be seen, opposite to the improved SA algorithm in this article, the RPROP algorithm has long training time, and also is not easy to convergent. The final compression results compressed by the RPROP algorithm are unsatisfactory either.

Table 1. Compression results based on RPROP algorithm and improved SA algorithm

Algorithm		PSNR	SNR
RPROP algorithm	8 times	76.23	17.50
	16 times	64.94	7.03
	32 times	58.20	2.30
	64 times	55.33	2.05
Improved SA algorithm	8 times	97.41	38.60
	16 times	106.41	47.60
	32 times	104.36	45.55
	64 times	103.24	44.43

Fig. 5. Original image

(a)



(b)

(c)

Fig. 6. Reconstruction image based on RPROP algorithm
(a) 8 times of image compression
(b) 16 times of image compression
(c) 32 times of image compression

(a)

(b) (c) (d)

Fig. 7. Reconstruction image based on improved SA algorithm
(a) 8 times of image compression
(b) 16 times of image compression
(c) 32 times of image compression
(d) 64 times of image compression

4. CONCLUSION

This article proposes an improved algorithm aiming at the disadvantages of SA algorithm and RPROP algorithm. The improved SA algorithm adjusts the learning rate and the momentum factor locally at the same time. This arithmetic performs the simulation experiment to curve fitting. For the improved SA algorithm, the testing results show that the convergence speed is quicker and the oscillation phenomenon is improved distinctly. Then, we use this improved method in the image compression and make a contrast between the SA algorithm improved and the RPROP algorithm.

5. REFERENCES

[1] D.E. Rumelhart, “Learning internal representations by back-propagating errors”, Nature, 1986, 533-536.
[2] J. Jiang, “Image compression with neural networks-A survey”, Signal Processing: Image Communication, Vol.14, 1999, 737-760.
[3] W. Schiffmann, M. Joost, “Optimization of the back-propagation algorithm for training multilayer perceptrons”, Technical report, University of Koblenz , Institute of Physics,Rheinau1, Koblenz,Germany, 1994.
[4] B. Li, G. Li, et al, “Improving the SA learning algorithm”, Signal Processing, 2005, 21（6）：615-620.
[5] Koen Denecher, Dimitri Van De Ville et al, “Design of an improved lossless halftone image compression codec”, Signal Processing: Image Communication, Vol.17, 2002, 277-292.
[6] Robert Cierniak, Leszek Rutkowski, “On image compression by competitive neural networks and optimal linear predictors”, Signal Processing: Image Communication, Vol.15, 2000, 559-565.
[7] S. Costa, S. Fiori, “Image compression using principal component neural networks”, Image and Vision Computing, Vol. 19, 2001, 649-668.
[8] Y.L. Huang, R.F. Chang, “Error concealment using adaptive multilayer perceptrons (MLPs) for block-based image coding”, Neural Computer & Application, Vol.9, 2000, 83-92.
[9] L.L. Fen, Z.H. Yu, et al, “The research of neural networks in image compression”, Journal of North China Electric Power University, 2002, 29（2）：91-93.
[10] H.X. Mei, T.Z. De, “The simulation research of image compression based on neural networks of multilayer feedforward”, Computer Simulation, Vol.8, 2005, 118-121.

Registration for Stereo Vision-based Augmented Reality Using Projection Technique

Yan Wang, Yaolin Gu
School of Information Engineering, Southern Yangtze University
Wuxi, 214122, Jiangsu, P.R. China
Email: trebleflowers@163.com

ABSTRACT

Registration is the key problem in AR application systems. In this paper, a simple registration method is proposed using projection technique. It is composed of four procedures: detecting positions of markers and feature points, tracking of features using projection technique, calculation of the model-view matrix, and rendering the virtual objects. In the initial process, four points are specified to build the world coordinate system on which the virtual objects will be superimposed. In the registration process, projection technique is used to track the positions of feature points and calculate the model-view transformation for augmentation. This method can provide accuracy needed for general indoor augmented reality systems and it is performed in real time.

Keywords: Augmented reality, registration, projection technique, tracking.

1. INTRODUCTION

Augmented reality (AR) attempts to integrate computer-generated virtual objects and other information onto the images of the real scene, thereby achieves the enhancement of the reality. AR systems have three properties: combing real and virtual objects in a real environment; running interactively in real time; and registering (aligning) real and virtual objects with each other [1]. AR has received a great deal of attention as a new method in many application domains, such as the medical domain, education training, manufacturing, and entertainment. A number of applications have already been proposed and demonstrated [2, 3, 4].

To implement an augmented reality system, some problems must be solved especially registration because virtual objects should be superimposed on the right place as if they really exist in the real world. Now, registration methods are mainly based on sensor and computer vision. Sensor-based registration methods use sensors to track and measure the positions and orientations of objects in AR systems. The sensor tracking systems commonly include electromagnetic, optic, ultrasonic, and mechanic tracking systems. All of these tracking systems have some disadvantages respectively: the magnetic tracking is easily affected by the magnetic field and metallic substances in the working environment; the optic tracking is often faced with the problem of occlusion; and in the ultrasonic tracking systems, besides the occlusion , some other elements must be considered such as the environment noise, temperature, humidity and etc.; the mechanic tracking is not easily affected by the environment, but it is often constrained by its small working range. The high cost is also the common problem of sensor-based registration methods. However, computer vision-based registration approaches provide the potential for accurate registration without the need for any additional costly sensors. This technique can estimate the position and orientation of user's viewpoint from images captured by cameras.

In this paper, we propose a simple registration method for stereo vision-based AR, using projection technique. The method reported in this paper is able to provide accurate registration for general indoor AR systems. Section 2 presents the fundamental principle of registration. Section 3 presents in detail the proposed registration method by using projection technique. And finally, in section 4, conclusions are given.

2. FUNDAMENTAL PRINCIPLE OF REGISTRATION

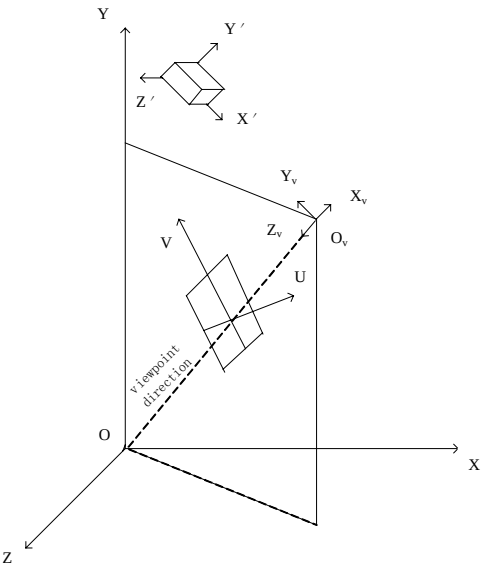


Fig. 1. Fundamental principle of registration

Generally, four coordinate systems are preferred to describe the registration principle. As shown in Fig. 1., XYZ is the world coordinate system, which is also called the real space coordinate system. $X'Y'Z'$ stands for the virtual space coordinate system, which is used to be the geometric description of the virtual objects superimposed. $X_V Y_V Z_V$ indicates the camera coordinate system which is also named the view space coordinate system. UV is a 2D coordinate system, representing the planar coordinate system of the projective images. The $O_V Z_V$ axis should be superposed with the viewer's viewpoint direction.

Because the orientation of the virtual objects to be superimposed in the real space is determined by the functions of AR systems, that is to say, the relationship between the virtual space coordinate system $X'Y'Z'$ and the world coordinate system XYZ is known, the translation from the geometric description of 3D virtual objects (x', y', z') in the virtual space coordinate

system to (x, y, z) in the world coordinate system can be represented as follows:

$$[x \quad y \quad z \quad 1] = [x' \quad y' \quad z' \quad 1]S, \qquad (1)$$

where S is the translation matrix between the virtual space and world coordinate systems. If the model-view matrix T between the world and camera coordinate systems is obtained, the geometric description of 3D virtual objects (x', y', z') in the virtual space coordinate system can then be transformed to (x_v, y_v, z_v) in the camera coordinate system as follows:

$$[x_v \quad y_v \quad z_v \quad 1] = [x' \quad y' \quad z' \quad 1]ST. \qquad (2)$$

After that, project the virtual objects in the camera coordinate system onto the image planar coordinate system UV, and then the registration of AR system is accomplished.

According to the analysis above, the registration of AR system mainly includes two steps:

1. Tracking the position and orientation of user's viewpoint.
2. Locating and rendering the virtual objects in the real space.

3. OUR REGISTRATION METHOD

We assume in this research that a pair of stereo cameras is virtually located at viewer's two eyes in an augmented reality system. And the camera intrinsic parameters are thought to be known in advance. Fig. 2. illustrates the flowchart of the proposed method. First, the markers and feature points are detected from a pair of stereo images using ARToolkit [5] (A in Fig. 2.). Then the feature points will be tracked in the subsequent images using projection technique (B in Fig. 2.). Next, the tracking results are used to estimate the model-view matrix which represents the relationship between the world and virtual coordinate systems (C in Fig. 2.). Finally, the virtual objects can be rendered using OpenGL (D in Fig. 2.).

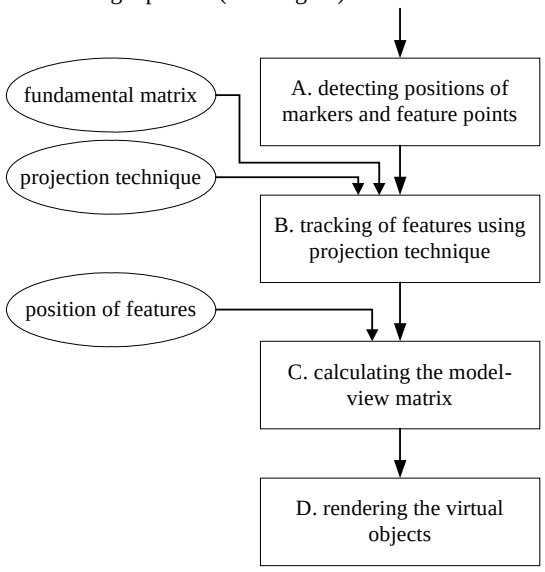


Fig. 2. Flow diagram of the registration method

3.1 Detecting positions of markers and feature points

In the first frame, markers and natural features are detected from a pair of stereo images. In our AR system, the marker detection method in ARToolkit is used to find the related markers and the feature points. These features will be the reference points in the whole augmentation process. Next, in order to insert the world coordinate system where the virtual object will be superimposed, four points in each of the two control images are specified respectively. These four points will form an approximate square and the origin of the world coordinate system will be the center of it.

3.2 Tracking of features using projection technique

In this section, the fundamental matrix and projection technique are used to track the markers and feature points. It is well known that there is a geometric constraint in the correspondence problem between two images, which is the epipolar geometry [6]. The fundamental matrix is used to indicate the epipolar geometry of the two images. In this research, 8-point method [7] is used to compute the fundamental matrix.

As section 3.1 points out, four points are specified respectively in two control images as point matches. Each point specified from the left image is matched with a corresponding point in the right image using projection technique with the epipolar constraint. When four point matches have been specified, the related projective 3D coordinates of specified square points are calculated to determine their locations in the other images in the video sequence. These projective 3D points will be further used in the tracking process using the reprojection technique.

Now, let F be the fundamental matrix between two images. There are numberless projection bases that satisfy the epipolar geometry. One possibility is to factor F as a product of an antisymmetric matrix $[e']_x$ and a matrix M, i.e., $F = [e']_x M$. As a matter of fact, e' is the epipole in the second image. Then, two projective camera matrices P and P' can be obtained as follows [8]:

$$P = [I \mid O], P' = [M \mid e']. \qquad (3)$$

For each image, its corresponding projective matrix can be computed, using the projective camera matrices and their corresponding 3D projective points. Hence, if a 3D projective point is got at the initial process using two images, the estimated matrix can be used to calculate its projection using the linear least square method during the tracking process. Therefore, the image projections of 3D projective points can be obtained from the updated projective matrix during the tracking process.

3.3 Calculating the model-view matrix

A model-view matrix that represents the relationship between the world and camera coordinate systems is determined using the image points obtained in section 3.2. The method in [5] is used to compute the model-view matrix.

3.4 Rendering the virtual objects

Since the camera intrinsic matrix is known in advance, using the model-view matrix computed in section 3.4, graphics

rendering procedures can be implemented by using OpenGL. The virtual object will then be aligned with the video sequence.

(a)

(b)

Fig. 3. Examples of registration from different camera positions. (a) and (b) are the 65th and 276th images in the whole video sequence, respectively.

The proposed registration method has been implemented by using Visual C++. Fig. 3. illustrates the registration results in one of our experiments. The marks '+' in the four corners of the white paper represent positions of the four points that we specified. The virtual object can be augmented on a world coordinate system formed by these four points. The indexed numbers ("0"-"7") indicate positions of the natural features and markers. In this experiment, a reading lamp as a virtual object is drawn on the table by using the estimated model-view matrix. (a) and (b) in Fig. 3. are the two images obtained from different camera positions. It can be observed in Fig. 3.: a reading lamp is registered stably even if the camera's position is changed.

4. CONCLUSIONS

In this paper, we propose a stereo vision-based augmented reality system based on the projection technique, and it is useful for the augmented simulation between the views of the real space and the virtual space. This registration method can be used in general indoor AR systems. In future work, we will attempt to improve the 3D interaction between human and computer, and human and environment.

5. REFERENCES

[1] R. Azuma, Y. Baillot, et al, "Recent Advances in Augmented Reality," IEEE Computer Graphics and Applications, Nov./Dec. 2001，21(6)：34-47.

[2] M. Rosenthal, A. State, et al, "Augmented Reality Guidance for Needle Biopsies: An Initial Randomized, Controlled Trial in Phantoms," Medical Image Analysis, 2002,6(3) ： 313-320.

[3] B. MacIntyre, M. Lohse, et al, "Integrating 2-D Video Actors into 3-D Augmented-Reality Systems," Presence, 2002， 11（2）： 189-200.

[4] W. Barfield, K. Baird, et al, "Applications of Wearable Computers and Augmented Reality to Manufacturing," Fundamentals of Wearable Computers and Augmented Reality, W. Barfield and T. Caudell, eds., Lawrence Erlbaum, 2001, 695-724.

[5] H. Kato, M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System," Proc. Second IEEE and ACM Int'l Workshop Augmented Reality, 1999, 85-94.

[6] Z. Zhang, R. Deriche, et al, "A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry," Artificial Intelligence J., 1995,vol. 78, 87-119.

[7] R.I. Hartley, "In Defense of the Eight-Point Algorithm," IEEE Trans. Pattern Analysis and Machine Intelligence, June 1997, vol. 19（6）, 580-593.

[8] R.I. Hartley, "Projective Reconstruction and Invariant from Multiple Images," IEEE Trans. Pattern Analysis and Machine Intelligence, Oct. 1999, 16（10）: 1036-1040.

A Dual Secure Digital Image Watermark Algorithm Using Wavelet Fractal And Chaotic Sequence

Sumin YANG, Jiazhen Wang, De-yun Peng, Zhengbao Zhang
Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China
Email: ysmyxh@tom.com

ABSTRACT

A new digital watermark algorithm is presented in this paper, on the one hand which utilizes the character of wavelet fractal coding to resist geometrical transformations, on the other hand which selects real hiding points from hiding candidates randomly according to a chaotic sequence, which is sensitive to initial values and random alike. So the robustness of the algorithm is ensured by two ways, and it is hardly possible to get watermark through unlimited access to the test image because of little mutual information. The experiments prove that the algorithm can resist some kinds of transformations such as clipping, JPEG compression and other geometrical transform, and the performance is better than the zero-tree method.

Keywords: Digital Watermark, Wavelet Fractal Coding, HVS, Chaotic Sequence.

1. INTRODUCTION

Watermarking has become an essential tool for multimedia copyright protection. Numerous techniques have been proposed for watermarking audio and image contents [1, 6]. For copyright protection purposes, the watermark should remain in the host image regardless of any reasonable processing that it may undergo. Most current watermarking algorithms are based on the zero-trees of wavelet which transforms the host image in the wavelet domain, and embed the watermark by changing the parity of the zero-tree coefficients or setting the wavelet coefficients different integer [5]. But the ability that zero-trees resist the various geometrical transformations and JPEG compression is relatively bad. However, at the present the serious problem is that many attacks have been proposed to remove watermark from the host image or at least render it undetectable. These attacks take advantage of the nonempty overlap between the permissible domains for image modification. The larger the overlap domain is, the easier the successful attacks become. If the pirate has unlimited access to the test image as a black box with the mutual information between watermarked pieces, they can estimate the watermark, so the hidden message can not only be easily destroyed, but also let out the secret of the message, and therefore decreasing the overlap between original image and watermarked image is most important.

In this article, we propose an alternative algorithm that combines wavelet fractal with chaotic sequence to avoid the mentioned problems, because the algorithm adequately utilizes the character that fractal coding to resist the geometrical transformations and the character of chaotic sequence that is sensitive to initial values and random alike, by chaotic sequence the real hiding points are selected from hiding candidates randomly. Then the embedding intensity of watermark depends on the HVS (Human Visual System) of host image, which ensures the imperceptibility of watermark. So the robustness and imperceptibility are enhanced.

The paper is organized as follows. In section 2, we discuss the relevant principle of wavelet fractal, zero-tree theory and chaotic sequence briefly. In section 3, we propose our new scheme and introduce the method of embedding, extraction and detection of watermark in detail, the experimental results to assess the performance are presented in section 4, and some characteristics and discussions are highlighted, finally concluding remarks are given in section 5.

2. THE PRINCIPLE OF WAVELET FRACTAL, ZERO-TREE AND CHAOTIC THEORY

2.1 Wavelet Zero-tree Theory

The concept of the zero-tree is firstly introduced in the EZW by Shapiro, the following is its definition: after the wavelet transformation is performed on the image, if the wavelet coefficient is less than the threshold T, we can consider that the coefficient is not important relative to the threshold, if the coefficient in the coarse scale is not important relative to the threshold, and all the coefficients in the finer scale in the same direction are not important, these coefficients will constitute the zero-trees.

In the zero-tree algorithm, because the coefficients in

$$\begin{aligned} X_{i,j} &= -\alpha \quad when \quad \omega_i = -1 \\ X_{i,j} &= \alpha \quad when \quad \omega_i = 1 \end{aligned} \tag{1}$$

these zero-trees are less effect on the image, we can search all the zero-trees in the wavelet domain, and embed the watermark in them, the formula is as follows :

where $X_{i,j}$ is the coefficient, ω_i is the embedded watermark, α is a scale integer.

By the method we can embed the watermark in the image, and can easily detect it. Because of the small number of the zero-trees, the amount of the embedded watermark is limited. Moreover, for the image having complex texture and boundary, the coefficients in the coarse scale are near zero, but the coefficients in the finer scale are non-zero. On the basis of the zero-tree principle, if all the coefficients in the zero-tree domain are adjusted by the same way, the perceptible quality of the image must be affected, but with the new method of the paper ,we can solve the previous problem.

2.2 Wavelet Fractal Theory

The foundation of wavelet fractal approach is the similarity of wavelet zero-trees. The wavelet coefficients can be classified as Domain trees (wavelet parent-trees) and Range trees (wavelet sub-trees). According to the fractal theory[2,4], the wavelet parent-trees are classified as similar and non-similar domain trees by the matching algorithm, so coefficients in the finer scale can be iterated by the

coefficients in the coarse scale, which make us get original integrated image even if the image has been destroyed.

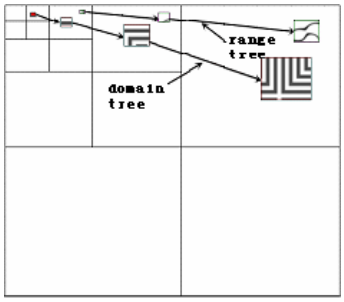


Fig. 1. Wavelet tree's construction

In the wavelet domain, the wavelet sub-tree denotes the wavelet trees at the beginning from the $l-1$ level ,the wavelet parent tree denotes the wavelet trees at the beginning from the l level ,the figure 1 is the sketch map

$$T(R_i) = \mu * B(S(D_i)) \quad (2)$$

of the wavelet tree construction. The matching formula is as follows:

where $S(D_i)$ denotes the coefficients of higher frequency in the wavelet parent tree D which make the number of range tree same as the number of the domain tree; B is the eight transformations of rotation and reflection ; μ is scalar factor.

According to the formula (2), we can get the matching similar domain trees, which is the basis of the iteration.

2.3 Chaotic Sequence

The chaotic sequence has the character that is sensitive to initial values and random alike. By lots of analysis, we use the Arnold chaotic sequence [3] having periodicity, which make us get the original image at one time. We can apply this character to permute the image so as to encrypt the message and get the original message, which overcomes the pitfall that random-permutation can't be gotten back, the following is the formula:

where N is the width and height of image, and $(x', y'), (x, y)$ is the position of image pixel after and before being transformed.

Fig. 2. shows the result being applied the Arnold transformation on watermark, we can see that the original

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \quad (3)$$

message has been ambiguous, which improves the secret extent of message and lengthens the time of decrypted.

Fig a original watermark Fig.b frequency=16 Fig.c

Fig. 2. The Contrast of Arnold Transformation

3. THE DUAL SECURE DIGITAL IMAGE WATERMARK ALGORITHM

3.1 Outline Of the Scheme

The wavelet transformation is performed on the host image. Then we can divide the wavelet zero-trees from all wavelet trees absolutely by comparing the error with the various level thresholds, then we utilize the formula (2) to search the matching fractal domain trees, then domains where error is smaller than the threshold value are selected to embed the watermark.

To increase the difficulty of decoding the watermark, the position of embedding and extraction of the watermark must do by a key. Firstly, generate a matrix of 0/1 with the same size of host image; secondly, transform the matrix by Arnold transform algorithm under the control of the bit sequence of the key; thirdly, use the transformed matrix to control the embedding depth of the watermark. .

3.2 Watermark Generation

The scheme that the watermark message is generated is as follows:

- The original message (m×m) is transformed by the Arnold chaotic sequence. The sequence is defined as $\omega_{m_1}, \omega_{m_1}, \Lambda, \omega_{m_{m\times m}}$.
- A pseudo random sequence of ±1 (same size as the host image) is generated for each bit of the payload to be embedded, and in order to remain little interference between the sequences, the correlation between the

$$\omega_{m_i} = \begin{Bmatrix} \omega_{ri} & m_i = 1 \\ -\omega_{ri} & m_i = -1 \end{Bmatrix} \quad \omega_i = \frac{\sum_i \omega_{m_i}}{\delta(\omega_{m_i})} \quad (4)$$

  sequences should be orthogonalized. The sequences are defined as ω_{ri} $(i = 1,2\Lambda, m\times m)$.
- The message template is generated by the following formula :

  Where $\delta(\omega_{m_i})$ is sample variance?
- We can impose the DWT transforming on the ω_i, Then by the perceptual model ,we can scale ω_{m_i} in order to remain the imperceptive character of host image.

3.3 Watermark Embedding

The watermark that can be embedded is generated according as section 3.2. we can embedded it into the wavelet fractal trees, the process is as follows:

- Four-staged wavelet transform is performed on initial host image.
- Search all fractal trees in the wavelet domain, and compute the error.
- Generate a 0/1 matrix of the same size to the image. Transform the matrix by Arnold transform algorithm

$$X^*_{i,j} = X_{i,j} + \alpha\omega_i \quad (5)$$

  iteratively under the control of the key's bit sequence. The result is a control matrix M . If the element of M is 1 and the position belongs to hiding candidates, the scaled watermark is added on the wavelet coefficients, else the scaled watermark is subtracted on the wavelet coefficients. The embedding formula in fractal domains is as follows :

  Where $X^*_{i,j}$, $X_{i,j}$ denote the wavelet coefficient after and before being embedded watermark respectively, α is scalar factor, ω_i is embedding watermark.

3.4 Watermark Extraction and Detection

1) Watermark Extraction

At first, we perform same scale wavelet transformation on the test image. According to the defined threshold, all the fractal wavelet trees are searched. Then depending on the fractal arguments that are saved in section 2.2, we calculate fractal iteration to get the coefficients $X'_{i,j}$ in the finer scale.

At last the embedded watermark can be gotten as following rule:

- if $X'_{i,j} > 0$ and $X^*_{i,j} > X'_{i,j}$ then $\omega^*_i = 1$
- if $X'_{i,j} > 0$ and $X^*_{i,j} < X'_{i,j}$ then $\omega^*_i = -1$
- if $X'_{i,j} < 0$ and $X^*_{i,j} > X'_{i,j}$ then $\omega^*_i = 1$
- if $X'_{i,j} < 0$ and $X^*_{i,j} < X'_{i,j}$ then $\omega^*_i = -1$

Where $X'_{i,j}$ is the fractal iterated coefficient, $X^*_{i,j}$ is the coefficient of test image, ω^*_i is the extracted watermark.

2) Watermark Detection

After extracting the watermark, we can estimate if there is watermark by the correlation of both ω^*_i and ω_i, the formula is as follows:

$$\rho(\omega,\omega^*) = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\omega_{(i,j)} * \omega^*_{(i,j)}}{(\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\omega_{(i,j)} * \omega_{(i,j)})^{\frac{1}{2}}) * (\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\omega^*_{(i,j)} * \omega^*_{(i,j)})^{\frac{1}{2}})} \quad (6)$$

With the comparative result between $\rho(\omega,\omega^*)$ and $\rho_{threshold}$ (given threshold), we can judge whether there is watermark in test image or not.

4 EXPERIMENT RESULTS

In order to perform fractal action, we must select the wavelet transformation that has symmetry or dissymmetry; we choose Daubechies 7/9 wavelet. The host image is 512×512, known as lena.bmp, the watermark is 32×32 binary grayscale image. The following Fig. 3. and Fig. 4. are the results after we perform our algorithm.

We can see that there is hardly any change between the original image and embedded image. Fig. 5. and Fig. 6. are the comparative results when performing JPEG transformation and cropping on the test image with algorithm proposed in the paper and zero-tree algorithm. From the Fig. 5. and Fig. 6. , we can see that the BER (Bit Error Rate) is decreased obviously.

In order to show the effect of the algorithm, we contrast its PSNR with zero-tree's, the result is shown at the following Table 1. From the table 1, we can see that the PSNR of our algorithm is higher than the zero-tree's; we believe that the above-mentioned results can be further improved through a few modifications to the parameters of wavelet fractal and chaotic sequence.

Table 1. The Comparative Result of two algorithm

Embedding watermark algorithm	**PSNR(db)**
Zero-tree	39. 62
Our algorithm	39. 99

Fig. 3. Original Watermark and

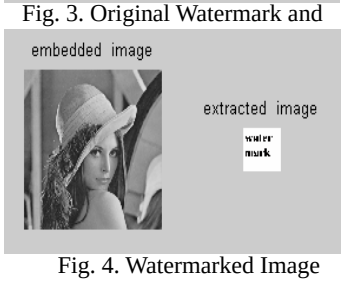


Fig. 4. Watermarked Image and Extracted Watermark

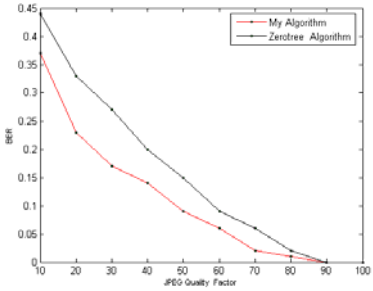


Fig . 5. The Comparison of JPEG

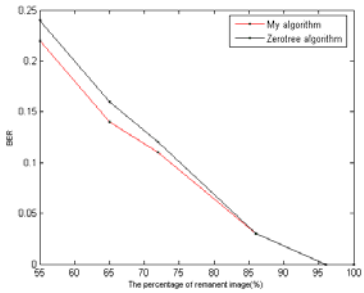


Fig. 6. The Comparison of Cropping

5. CONCLUSION

In this paper, we propose a novel robust digital watermark algorithm that incorporates wavelet fractal and chaotic sequence, the proposed technique is ensured security from two ways, so it is highly robust in compression, cropping and other transformations. In fact, the watermark can be extracted fairly accurately even if the images are almost completely destroyed by fractal iteration. Future work will concentrate on modifying the method more practical so as to apply it to engineering work.

6. REFERENCES

[1] H. Inoue, A. Miyazaki, et al, "A Digital Watermark Based on the Wavelet Transform and Its Robustness on Image Compression"[C], International Conference on Image Processing (Icip'98), Oct. 4-7 , 1998, 2:391-395.

[2] Cardinal, "Fast Fractal Compression of Greyscale Images"[J]. IEEE Transactions on Image Processing,

2001, 10(1): 56-67.

[3] DongXu Qi Fractal and Its Application, Beijing: Science Press, 1994.143-145(in Chinese).

[4] J. Li, C.Kuo, “Image Compression with Hybrid Wavelet-Fractal Coder”, IEEE Transaction on Image Processing, 1999, 8(6): 868-874.

[5] K. t. Lo, X. D. Zhang, et al, “Universal Perceptual Weighted Zerotree Coding for Image and Video Compression”[J], IEEE Proceeding of Image, vision and Signal Processing, 2000, 147(3): 124-127.

[6] S. Voloshynovskiy, T. Pun, “Capacity-Security Analysis of Data Hiding Technologies”, IEEE International Conference on Multimedia and Expo, 2002, 8(26-29), vol. 2: 4.

Study on Image Edge Detection Based on Set Pair Analysis

YiAn Liu XianHai Meng Yuan Liu ShiTong Wang
School of Information Engineering, Southern Yangtze University,
Wuxi 214122, P.R. China
Email: lya_wx@yahoo.com.cn

ABSTRACT

To solve the edge detection problems in image processing, a new edge detection approach based on SPA is proposed in this paper. In the proposed approach, the eight natural adjacent pixels which lie in horizontal, vertical and two diagonal directions to the pixels to be detected are organized as the four set pairs in a two-by-two way. The gray level discontinuity between the objects and background and the similarity of the pixels in non-edge areas are also taken into account. Meanwhile, in order to extract the image edge adaptively, the global identical degree of the image is used as the threshold. If the minimal identical degree of the four set pairs adjacent to the pixels to be detected is greater than the threshold, we obtain the criterion used to judge the detected point to be an edge pixel. The experimental results indicated that this approach is not only simpler but also has better performance than the traditional approaches in such aspects as anti-jamming, detectable precision enhancement and edge-details protection.

Keywords: edge detection; set pair analysis; degree of identity**;** degree of opposition; gray level.

1. INTRODUCTION

Edge detection is one of the classical topics in image processing. Because the basic shape and primary characteristic information of the image can be determined from the edge, edge detection plays a pivotal role in image processing. Its task is to determine and extract the edge information of image, so as to achieve the preprocessing goal[1] for image analysis, target identification and image encoding. Therefore, the high or low quality of the edge detection will greatly influence the results of image post-processing.

The recent researches on edge detection have obtained great achievements. Classical approaches, such as Sobel, Prewitt, Canny and Laplacian etc operators[2], usually used the discontinuity of the edge gray level to detect the edges by the rules of the pixel's first or second derivatives in little sub domains. Because of the sensitivity of these algorithms to the noise, these approaches perhaps not only enlarged the noise but introduced blurs on the resulted edges as well as discontinuity in some pixels so that the obtained visual effects are poor. Therefore, besides current improvements for the traditional approaches[3,4,5], new detection theories, such as mathematics morphology[1], wavelet transformation[6], rough set[7],gray system[8] are being incorporated into these approaches.

In this paper, the image edge detection problem is regarded as an uncertainty system. According to Set Pair Analysis, the eight natural adjacent points which lie in horizontal, vertical and two diagonal directions to the pixels to be detected may be organized into four pairs in a two-by-two way. We analyze and research quantitatively the IDC (Identical-Discrepancy-Contrary) connection and the transformation from the positive-negative and the description of uncertainty methods between set pairs to obtain the set pair connection degree formula: $\mu = a + bi + cj$ which means each direction of the neighborhood to the pixels needs to be detected. Then we use the edge gray level discontinuity between object and background and the similarity of the pixels in non-edge areas to extract the parts of image edge. Meanwhile, the paper uses the global identical degree between object and background and the minimal identical degree of the pixel to be detected in the neighborhood to design an adaptive threshold, to judge whether a pixel to be detected is an edge pixel or not. Compared with the traditional edge detection algorithms, this approach here is not only simpler but also has better performance in such aspects as anti-jamming, enhancing detection precision and protecting edge details. The experimental results also indicated that this approach is valid and feasible.

2. THE BASIC CONCEPT OF SET PAIR ANALYSIS

Set Pair Analysis (SPA)[9], as a new theory of system analysis, was presented by a Chinese Scholar Zhao Ke-qin in the late of 1980s. It deals with the IDC's quantitatively analysis of uncertainty system. Nowadays, it has been extensively applied in the fields of industry, agriculture, socio-economy and military, etc[10]. The so-called set pair refers to pairs which is composed by two relative sets. Its kernel idea is to regard the determination and indetermination as an uncertainty system, and then to analyze quantitatively the IDC and the transformation between set pair using the positive-negative and uncertainty principle, so that we can get set pair connection degree formula: $\mu = a + bi + cj$ under some problem conditions, and then extend it to the $m > 2$ cases, such that the relationship, prediction and control about the system can be further obtained. In this system, a, b, c must be satisfied with: $a + b + c = 1$, where a denotes the identity of characteristic between two sets, the degree of identity for short; c denotes the opposition of some characteristics, the degree of opposition for short; b denotes the discrepancy of characteristic between two sets, the degree of discrepancy for short. i, whose range is [-1, 1], is discrepancy tag or relevant coefficient. j, whose value is -1, is opposition tag or relevant coefficient.

3. DETERMINING THE CONNECTION DEGREE COEFFICIENT

In SPA, it is very vital to describe and measure the degree of opposition clearly because they have something to do with the description of IDC. According to [11], common opposite concepts could be divided into five types: reciprocal type,

existence-nonexistence type , positive-negative type, complementary type and false-true type，and reciprocal type is the most important form in the fields of science and technology. Therefore, our research on the reciprocal relationship between each set pair adjacent to the pixel we interested is based on the reciprocal types. In this paper, eight natural adjacent pixels which lie in horizontal, vertical and two diagonal directions to the pixels to be detected are organized as four set pairs in a two-by-two way and the two gray values are regarded as a data pair in each direction, because data pair is a special example of the set pair, it also has the same IDC relationship.

Suppose that the gray values s , t of both A and B pixels fall in $[0, L]$, L is the gray level of the image and $s \le t$. Then, according to [12], it is easy to know the proportion of s of the data pair (s,t) in the first order opposite interval $[\frac{1}{t}, t]$ of t :

$$p = \frac{s}{t - \frac{1}{t}} = \frac{st}{t^2 - 1}$$

where if $t = 1$, set $t = 2$, $s = s + 1$; if $s = t = 0$, set $s = t = 2$; if $s = 0$, $t \neq 0$,set $s = 2$, $t = t + 2$. Then, the opposite degree of s can be obtained in $[\frac{1}{t}, t]$, as follows:

$$c = 1 - p = \frac{t^2 - st - 1}{t^2 - 1} \qquad (1)$$

On the other hand，the identical degree of s and t is:

$$a = \frac{s}{t} \qquad (2)$$

So, the discrepant degree of (s,t) can be obtained as follows:

$$b = p - a = \frac{st}{t^2 - 1} - \frac{s}{t} = \frac{s}{t(t^2 - 1)} \qquad (3)$$

Hence, the connection degree formula of (s,t) can be described as the following expression:

$$\mu = a + bi + cj = \frac{s}{t} + \frac{s}{t(t^2 - 1)} i + \frac{t^2 - st - 1}{t^2 - 1} j \qquad (4)$$

Obviously, the values of a, b, c can reflect some state or possible trend of the interested object, such as sudden change, slow change or unchanging of pixel's gray level value.

4. EDGE DETECTION ALGORITHM

The basic characteristics of image edge pixels is to divide an image into different areas so that the pixels in each one have the same or similar characteristics[13]. Therefore, for the $M \times N$ image f , we should consider the following three cases, to judge whether a pixel $f(i, j)$ to be detected is an edge pixel or not.

① the pixel $f(i, j)$ is a background pixel or a target pixel;

② the pixel $f(i, j)$ is a noise pixel;

③ the pixel $f(i, j)$ is an edge pixel.

By means of the above three cases, we divide the pixels into two sets, namely A and B , based on the average gray level of the image: in A , all the pixels gray level are greater than the average gray level and in B , all the pixels gray level are less than the average gray level. Then the two average gray values of both set A and set B are considered as a set pair, and the formula (4) is used to compute the global identical degree a_0 of image f , and to take $\xi = a_0 + std$ as the threshold. Here std is the normalized standard deviation of the gray level for the image. If the minimal identical degree $a_{\min}$ of the four set pairs adjacent to the pixel to be detected satisfies $a_{\min} > \xi$, the pixel to be detected is regarded as the non-edge pixel, and conversely, we regard the pixel as the edge pixel. The reason is, when the eight natural adjacent pixels which respectively lie in horizontal, vertical and two diagonal directions to the pixels to be detected $f(i, j)$ are organized as the four set pairs, for the above case ①, since the gray values of pixels in the background or target areas vary slowly and have greater correlation between pixels, each of the four set pairs in the neighborhood of $f(i, j)$ can get greater identical degree. If the minimal identical degree $a_{\min}$ of the four set pairs satisfies $a_{\min} > \xi$, we may exclude the probability of that the pixel $f(i, j)$ is an edge pixel. For the above case ②, when the pixel $f(i, j)$ is a noisy pixel, it has less correlation with its adjacent pixels because of the randomness of noise. But the pixels adjacent to the noisy pixel still have greater correlation in the same areas, so that the minimal identical degree $a_{\min}$ of the four set pairs can be obtained and satisfies $a_{\min} > \xi$. Therefore, for the pixel $f(i, j)$, we can exclude the noisy pixel and achieve the goal which restrains its noise. For the above case ③, when the pixel $f(i, j)$ is an edge pixel, because the gray values of pixels along image edge direction vary gently while the gray values of pixels along the vertical or crossed direction of image edge vary intensely. So, in the four directions of the pixel $f(i, j)$, there is at least a set pair, minimal identical degree satisfies $a_{\min} \le \xi$. Therefore, the pixel $f(i, j)$ may be shown as an edge pixel. The concrete steps of the above algorithm can be described as follows:

Begin:

① For inputting gray level image f , we use the median filter first to suppress some noises. Because the median filter can filter a variety of noise as well as can protect well detail information of the image edge signal [14] to some extend.

② After the filtered gray level image, we still denote f , to compute its global identical degree a_0 and the std of normalized gray level standard

deviation, and set $\xi = a_0 + std$ for the threshold value.

③ For each pixel $f(i, j)$ of f, its eight natural adjacent pixels are composed of four set pairs in terms of horizontal, vertical and two diagonal directions, i.e. $(f(i, j-1), f(i, j+1))$, $(f(i-1, j), f(i+1, j))$, $(f(i-1, j-1), f(i+1, j+1))$ and $(f(i+1, j-1), f(i-1, j+1))$. Here, in order to compute them conveniently, set $i = 2,3,\cdots,M-1$; $j = 2,\cdots,N-1$. Because the influence of image circumferential points may be ignored.

④ By means of formula (4), we compute the identical degree a_k of four set pairs in turn, $k = 1,2,3,4$, and obtain the $a_{\min} = \min(a_1, a_2, a_3, a_4)$.

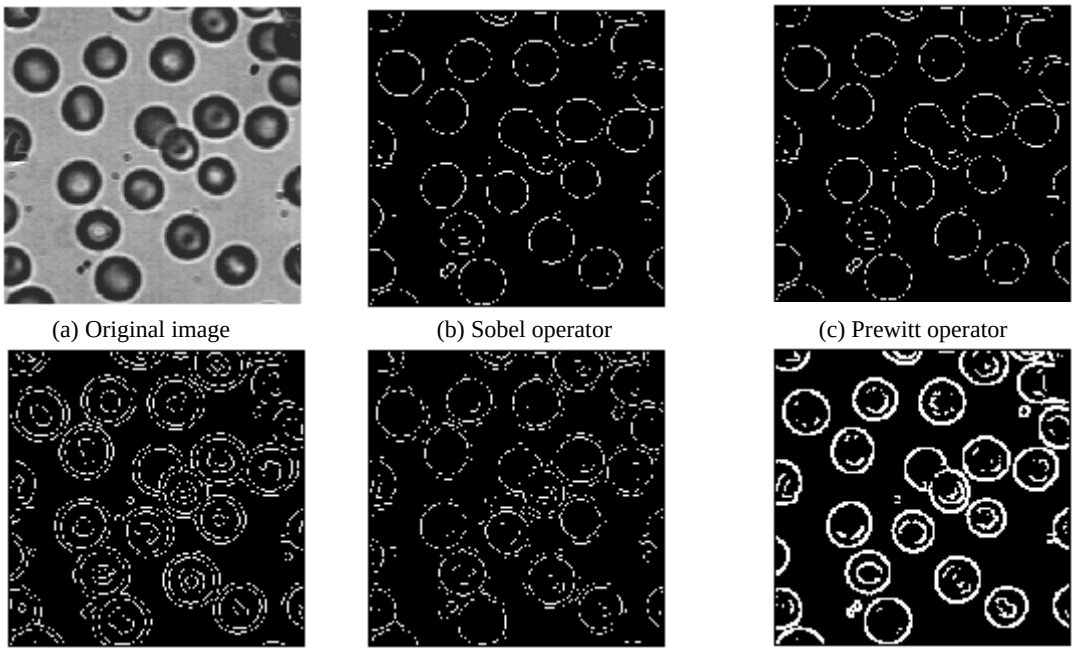
⑤ Judge whether $a_{\min} > \xi$ is true or not. If true, then the point $f(i, j)$ is non-edge point, otherwise it is an edge point.

⑥ Repeat the steps from ③ to ⑤, the image edge detection may be finished.

End

5. EXPERIMENTAL AND RESULTS

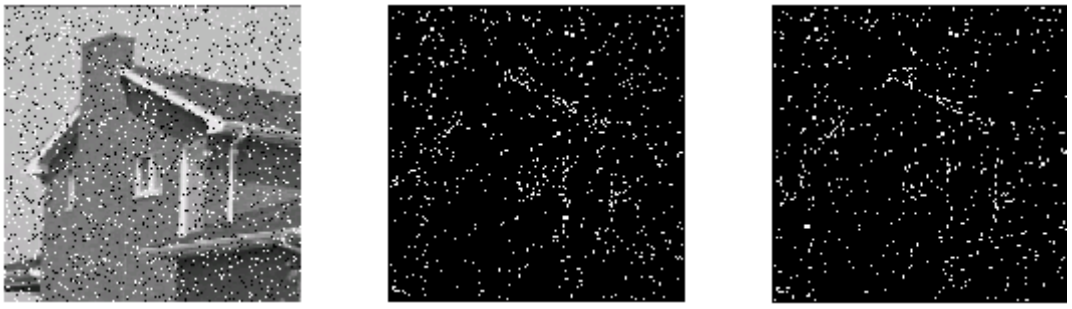
In order to prove the validity and the feasibility of SPA in image edge detection fields, the following experiments have given the compared results for Sobel, Prewitt, Canny, Laplacian, SPA approaches in Matlab 7.0 on our PC in the paper. Here, the first experiment dealt with the original image without noise, shown as Fig.1(a), and got the compared results; the second experiment added the salt & pepper noise to the original image, shown as Fig.2(a), and got the compared results



(a) Original image (b) Sobel operator (c) Prewitt operator

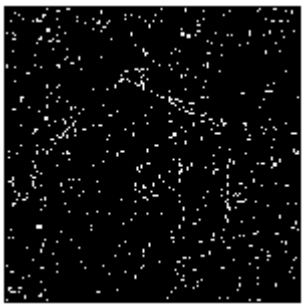
(d) Canny operator (e) Laplacian operator (f) SPA method

Fig. 1. The image of edge detection that does not add noise



(a) Original image (b) Sobel operator (c) Prewitt operator


(d) Canny operator


(e) Laplacian operator


(f) SPA method

Fig. 2. The image of edge detection that adds salt&pepper noise, and noise intensity is 0.02

As can be seen from the fig. shown above, the image edge was discontinuous when traditional detection algorithms were used. Specially, as the original image was added strong noise, the traditional detection algorithms ware nearly invalid. On the contrary, the SPA approach could get clear, continuous edges. It indicated that the SPA approach had strong ability against noise and better vision effects. On the other hand, from the time of image processing, the traditional algorithms of Sobel, Prewitt, Canny and Laplacian took 131 ms, 90 ms, 630 ms, 160 ms respectively in Fig.1 and 140 ms, 91 ms, 640 ms, 160 ms in Fig.2 while the SPA approach took 581 ms in Fig.1 and 591 ms in Fig.2. It also showed the SPA approach was faster than Canny algorithm, but we must still do our best to enhance the speed of edge detection so as to satisfy real-time image processing in the future.

6. CONCLUSION

Edge detection problem is not only important to image analysis, but also is the research foundation of pattern recognition and computer vision. In order to find a new detection approach, which has strong anti-jamming, high detection precision, good protection of edge details and good detection speed, edge detection has been a research focus on both home and abroad all the time. Maybe it is a good attempt that we use set pair analysis to deal with the edge detection problem in this paper. According to the experimental results, SPA algorithm is not only simple and effective, but has also better performance than traditional operators on anti-jamming, advance of detection precision and protecting edge details.

7. REFERENCES

[1] H. Chen, S.Z.Wang, et al, “Research based on mathematics morphology image edge examination method.” Journal of Engineering Graphics, Beijing, China, vol.2, no.2, pp.112-114, 2004.

[2] A Rosenfeld, A C Kak, Digital picture processing. Academic, New York, 1976.

[3] H Jeong, C I Kim, “Adaptive determination of filter scales for edge detection.” IEEE Transactions on Pattern Analysis Machine Intelligence, vol.14, no.5, pp.579-585, 1992.

[4] L.Wang, M.Y. Mo, et al, “An improved method of edge detection based on canny's theory.” Journal of Image and Graphics, Beijing, China, vol.1, no.7, pp.191-192, 1996.

[5] J.Wan, T.R.Xu, Study on image edge detection method based on laplacian operator[J]. Modern Electronic Technique, Beijing, China, vol.188, no.21, pp.92-93, 96, 2004.

[6] F.W.Chen, Wavelet analysis and its application on image processing. Science Technology & Publication, Beijing, China, 2002.

[7] D.P.Sui, Y.Zhou, et al, “An improved canny operator based on coarse computing”, Computer Engineering and Applications, Beijing, China, vol.41, no.9, pp.37-39, 2005.

[8] M.Ma, Y.Y.Fan, et al, “A novel algorithm of image edge detection based on gray system theory”, Journal of Image and Graphics, Beijing, China, vol.8, no.10, pp.1136-1139, 2003.

[9] K.Q.Zhao, Set Pair and Set Pair Analysis---A new concept and a new method of system analysis, Paper of System Theory and Region Designing Proseminar, BaoTou, China, August, 1989.

[10] K.Q.Zhao, Set pair analysis and its application [. Zhejiang Technology & Publication, HangZhou, China, 2003.

[11] K.Q.Zhao, “Opposition division, measurement and application based on set pair analysis.” Science Technology and Dialectics, Beijing, China, vol.11, no.2, pp.26-30, 1994.

[12] K.Q.Zhao, “Depiction and application based on the relationship of set pair analysis”. Nonferrous Metals Engineering & Research, Beijing, China, vol.17, no.3, pp.59-62, 1996.

[13] J.Qin, S.M.Zhou, A method of state power ordering on connection number [J]. Journal of Guangxi Normal University, Guangxi, Beijing, China, vol.21, no.3, pp.41-44, 2003.

[14] Z.Tan, F.M.Bao, et al, Digtal image fusion[M]. Publication of Xi'an Jiaotong University, Xian, China, 2004.

Three-dimensional Adaptive Meshing for Reverse Engineering

Lamei Yan1, Yueming Liu2, Jianbiao Xiao2
1. School of Mechanical Engineering, Hunan university of Technology,
ZhuZhou, Hunan, 412008,China
2. Yiyang Engineering Company,
(Hunan, Yiyang, 413000)
Email: y.lm@163.com

ABSTRACT

Adaptive meshing is an essential pre-requisite for various problems in finite element simulation. In this paper, we focus on the meshing of parametric surfaces conforming to a prescribed size field in reverse engineering. We propose a method that maintains a fine uniform grid in the important regions of the domain by using inexpensive action indicators to trigger selective refinement. The method is simple to code and adapt to existing finite element solvers. Our experimental results show that this scheme is extremely fast and scalable to large problems.

Key words: Mesh generation, Adaptive mesh computations mesh adaptation, adaptive meshing.

1. INTRODUCTION

By locally refining and de-refining the mesh either to capture flow-field phenomena of interest or to account for variations in errors, adaptive methods make standard computational methods more cost effective [1,2]. The efficient execution of such adaptive scientific simulations on parallel computers requires a periodic repartitioning of the underlying computational mesh. These repartitionings should minimize both the inter-processor communications incurred in the iterative mesh-based computation and the data redistribution costs required to balance the load. Recently developed schemes for computing repartitionings provide the user with only a limited control of the tradeoffs among these two objectives [3].

In this paper we present some new techniques for generating high quality triangular meshes. The methods are mainly based on fast algorithms for approximation and fairing. We use a static assignment for all patches to minimize communication overhead. The processor assignment for irregular patch objects uses a novel object ordering with cyclic mapping to achieve load balance even if computation associated with patch objects changes. Thus the generation, deformation and adaptation of suitable meshes is an important factor in our work. A key idea is to represent meshes with as few parameters as possible, while further successive refinements and deformations can be efficiently computed based on the knowledge of these parameters.

This paper is organized as follows. Section 2 gives distributed mesh representation. Section 3 gives an overview of our algorithm. Section 4 presents the experimental results. Finally, in section 5 gives some conclusions and future works.

2. DISTRIBUTED MESH REPRESENTATION

In 2-dimensions a Lepp based algorithm for the quality refinement of any triangulation mesh was introduced in [2], where the refinement of a target triangle to essentially means the repetitive longest-edge bisection of pairs rent Lepp (t_0), until the triangle to itself is refined. Lepp($\underline{t_0}$) is defined as the longest edge propagation path associated to $\underline{t_0}$ and corresponds to the sequence of increasing neighbor longest edge triangles that finishes when a terminal-edge is found in the mesh. For an illustration see Figure 1, where Lepp ($\underline{t_0}$)={t0, t_1, t_2, t_3} over the triangulation (a), and the associated terminal edge is the edge shared by triangles t_2, t_3. Triangulations (b) and (c) respectively illustrate the first and second refinement steps, while triangulation (d) corresponds to the final mesh when the Lepp Bisection procedure is applied to $\underline{t_0}$. Note that the new vertices were enumerated in the order they were created. The generalization of this algorithm to 3-dimensions mesh is formulated in the next section.

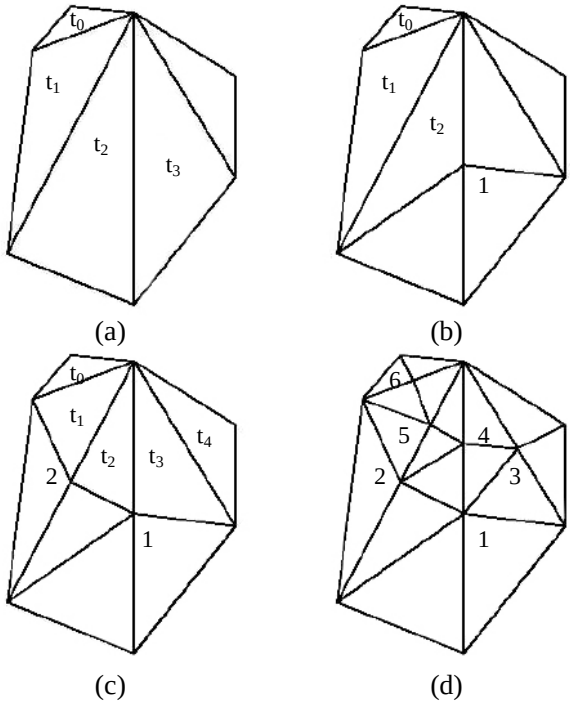


Fig .1. Lepp refinement of target triangle t_0 (a) initial mesh; (b) first refinement step; (c) second refinement step; (d) final mesh where triangle t_0 was refined

Mesh partitions have mesh entities in common. In Fig.2. we show an example of a distributed mesh with three partitions. We consider partition boundaries as artificial model entities that represent connections between partitions. The minimum mesh representation in parallel takes these new model entities into account: mesh entities equally classified on partition boundaries must be preset as well in the representation. This is a natural extension of

the serial definition.

In order to make partitions aware of remote entities that are present in other partitions, we use the algorithmic capabilities of AOMD [5]. For each mesh entity M_i^d classified on a partition boundary G_j^q, we send a message to each remote partition. This message contains

- The local address of M_i^d
- The list of M_i^d vertices iD's.

With vertices iD's, we are able to find the counterpart of M_i^d on every remote partition of G_j^q.

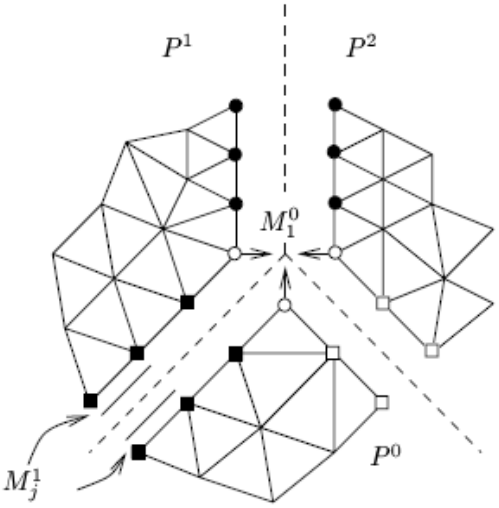


Fig. 2. Distributed mesh, three partitions P^1,P^2 and P^3. Vertex M_1^0 is common to all partition and is then classified to a partition vertex. On each partition, several mesh edges like M_j^1 are common to two partitions and so is classified to a partition edge separating two partitions.

3. OUR ALGORITHMS FOR THREE-DIMENSIONAL ADAPTIVE MESHING

3.1 Closed ball property

The restricted Voronoi diagram *Vs (P)* of a sampling *P* of a surface *S* has the closed ball property if the intersection of *S* with every Voronoi object in *V (P)* is homeomorphic to a closed ball whose dimension one smaller then that of the Voronoi object. Edselsbrunner and Shah [3] were able to relate the topology of the restricted Delaunay triangulation Ds (P) to the topology of S. As shown in Fig.3. Point x has six natural neighbors.

Theorem 1 *Let S be a surface and p be a sampling of s such that V_S (P) has the closed all property. Then D_S (P) and S are homeomorphic.*

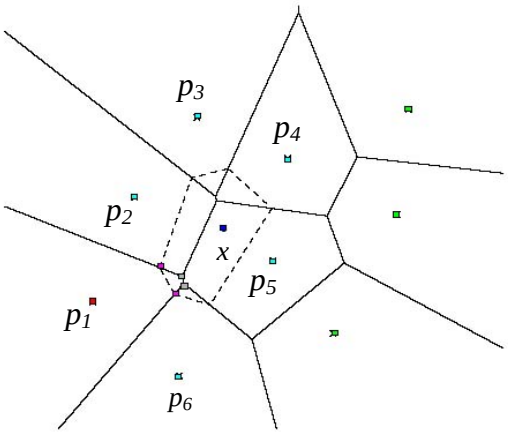


Fig. 3. Point x has six natural neighbors

3.2 Power diagram and regular triangulation

The concepts of Voronoi and Drlaunay diagrams are easily generalized to sets of weighted points. A weighted point o in R^3 is a topple *(z,r),*where $z \in R^3$ denotes the point itself and $r \in R$ its weight. Every weighted point gives rise to a distance function [6], namely the power distance function,

$$\pi_p : R^3 \to R, x \mapsto \|x - z\|^2 - r \qquad (1)$$

Let P be a set of weighted point in R^3.The power diagram of P is a decomposition of R^3 into the power cells of the points in P.

Theorem 2 The *natural coordinates satisfy the requirements of a coordinate system, namely,*

(1) for any $p \in P$, $\lambda_P(q) = \delta_{pq}$ where δ_{pq} is the Kronecker symbol and

(2) the point x is the weighted center of mass of its neighbors. That is,

$$x = \sum_{p \in P} \lambda_p(x)\, p, with \sum_{p \in P} \lambda_p(x) = 1. \qquad (2)$$

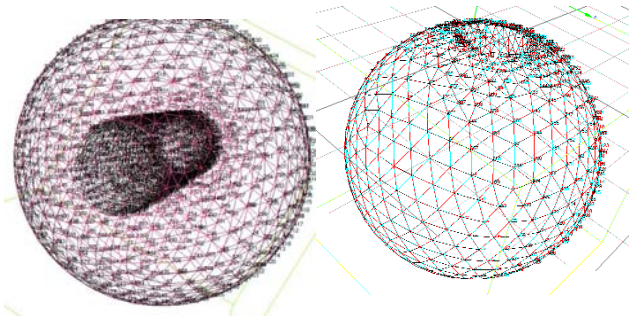
Induced distance function.

4. EMPIRICAL RESULTS

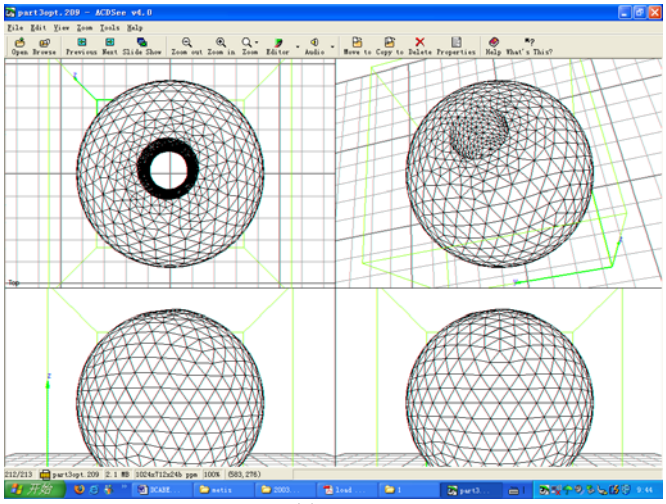
In this section, we present an example of mesh adaptation to demonstrate how to capture an analytically defined metric tensor. Unified Repartitioning Algorithm with optimized versions of scratch-remap (LMSR) [5] and multilevel diffusion (Wavefront Diffusion) [4] repartitioners.

Experimental Setup the experiments presented in this section was conducted on graphs derived from finite-element computations. For each graph, we modified the vertex and edge weights in order to simulate various types of repartitioning problems. Specifically, we constructed four repartitioning problems for each graph that simulate adaptive computations in which the work imbalance is distributed globally throughout the mesh. An example of an application in which this might occur is a

particle-in-mesh computation. Here, particles may be located anywhere within the mesh and may be free to move to any other regions of the mesh. The result is that both the densely and sparsely populated regions are likely to be distributed globally throughout the mesh. Fig. 4. (a). (b) shows the initial surface mesh and the adapted meshes at iterations 0 and 10. Fig. 4. (c) is the part.mesh use our method.



(a) (b)



(c)

Fig. 4. (a).(b). shows the initial surface mesh and the adapted meshes at iterations 0, 10 (c) is the part.mesh

5. CONCLUSIONS

In this paper, we have given a description of the surface of the domain of interest together with a metric map provided at the vertices of a background mesh. The problem is then to complete a mesh that conforms to this metric map. In addition to this (computational) metric map, surface re-meshing requires the construction of a discrete geometric metric (at the mesh vertices) map based on the intrinsic properties of the surface. Then, this metric is combined (intersected) with the given computational metric to define the actual adaptation metric. Load-balancing tools such as DRAMA [3] and Zoltan [5] can measure the times required to perform inter-processor communications and data redistribution for an application, use this information to automatically compute an accurate Relative Cost Factor, and then call URA with the correct value as the input. Therefore, the user need not determine a good value each time load balancing is required.

6. ACKNOWLEDGEMENTS

The work was supported by the science foundation of Hunan province ministry of finance and education No:06C101.

7. REFERENCES

[1] L. M. Yan, Y. W. Yuan, et al, A new three—dimensional shape reconstruction based on neural networks. ICEMI'2003. Vol (1), 81-84.

[2] P.J. Frey, Generation and adaptation of computational surface meshes from discrete anatomical data. Intenational journal for numerical methods in engineering. 2004(2), Vol (60): 1049-1074.

[3] H.Hoppe, T. DeRose, et al, "Piecewise Smooth Surface Reconstruction". Proceedings of SIGGRAPH,1994, 295 – 302.

[4] B. Curless, M. Levoy, "A volumetric method for building complex models from range images", in Proc. of SIGGRAPH'96, 1996, 303-312.

[5] P. G. Ciarlet, Basic error estimates for elliptic problems. Handbook of Numerical Analysis, Finite Element Methods (Part 1). North-Holland: Amsterdam, 1991 (2), Vol (2), 3477-352.

[6] H. Borouchaki, Hecht, et al, International Journal for Numerical Methods in Engineering; 1998(6); Vol (43), 1143~1157.

GPU in texture image processing

Zhipeng Xu
School of Physics Science and Information Engineering, Liaocheng University
Liaocheng, Shandong, 252059, China
Email: xu0510cn@yahoo.com.cn
And
Wenbo Xu
School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, 214122, China
Email: xwb@sytu.edu.cn

ABSTRACT

A GPU based approach has been developped to efficiently calculate texture features based on cooccurrence probabilities. The commomly used matrix based approach (the grey level cooccurrence matrix or GLCM) requires an unreasonable amount of computation, especially for image segmentation purposes. The GPU based approach calculates exactly the same results while significantly decreasing the time to calculate the texture features. This paper describes the implementation of a GPU based algorithm, demonstrates its applicability.

Keywords: GPU Computing, Co-occurrence Matrix, texture.

1. INTRODUCTION

Texture is an important part of image interpretation. A common approach to texture analysis uses grey level cooccurrence matrix(GLCM) texture features. GLCM estimates image properties related to second-order statistics. Each entry (*i*,*j*) in GLCM corresponds to the number of occurrences of the pair of gray levels *i* and *j* which are a distance *d* apart in original image. In order to estimate the similarity between different gray level co-occurrence matrices, Haralick [1] proposed 14 statistical features extracted from them. The co-occurrence method works well for a large variety of textures[2] , and gives good results in discrimination between textures. But the method requires huge memory, in order to decrease the co-occurrence matrix sizes, the number of gray-level should be set to 32 or 64, at the same time, the loss of gray-level accuracy is a resulting negative effect. Another disadvantage of the method is that it is computationally expensive. For a 512x512 pixels image, the time cost of co-occurrence matrices usually takes a few minutes, texture features then extracted from these matrices. So the method usually can not be used in real-time systems with common PC hardware.

In this paper, the GPU, which is graphics processing unit in common PC, is utilized to accelerate two features in the co-occurrence matrix method. The paper also gives an application in real-time systems.

2. GPU AND RELATED WORK

High performance graphics processing units (GPUs) are the calculating unit of modern graphics card. It plays the role of computation just like the CPU does in a computer. GPU has a powerful floating point capabilities and flexible programming interface. Due to their parallel nature, they are expected to be powerful tools in the fields of floating point calculations. For example, a Pentium 4 at 3GHz peaks out at 12 GFLOPS and has 5.96GB/s of memory bandwidth. By contrast, a Radeon X1800 XT can reach 83 GFLOPS and has 42GB/s of memory bandwidth.

GPU has been implemented in many scientific research, such as fast matrix multiplies[3], 3d convolution[4], morphological analysis[5], ray tracing[6], robot motion planning[7], sparse matrix solvers[8]. The fragment shader of a modern GPU can be seen as a stream processor. The texture image are expressed in vertices or texture. Then the vertex program or fragment program can play almost the same functions as CPU does. For example, the NV_fragment_program has a abundant instruction set, including texture fetching, add/sub/multiply/divide operations, etc. At the same time the appearance of floating point buffer (NV_float_buffer) makes it possible for the multi-pass rendering to be implementing in full precision..

3. TEXTURE FEATURES

Haralick [1] proposed 14 statistical features of co-occurrence matrix. In this paper, two features are discussed, the mean and contrast.

Texture features usually are extracted from a small sliding window moving horizontally from left to right. The size of window should be selected to accord with the texture. Heikkinen presents a fast algorithm of two texture features without calculation of co-occurrence matrix[9]. The algorithm can be described as follows: in Fig. 1., a window is moving from left to right, for two neighbouring windows, texture features can be calculated by updating the values computed in the previous window without using the co-occurrence matrix. The updating can be done purely using the gray tones appearing in the window taking into consideration only those pixel pairs that leave out and come into the window (cf. Fig. 1.). In the following formulation it is assumed that the displacement vector d has only the horizontal component, i.e. $d = (dx, 0)$.

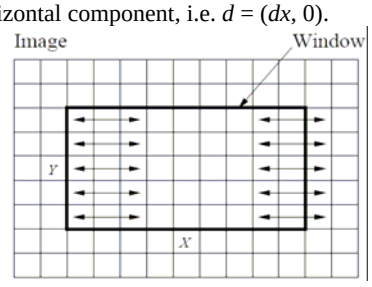


Fig. 1. Pixel pairs that leave out and come into the window that is moving from left to right, d = (2, 0).

$$m(t+1) = m(t) + \sum_{l=0}^{Y-1}\left(g_{X-d_x,l(t)} - g_{0,l(t)}\right) \quad (1)$$

$$c(t+1) = c(t) + \sum_{l=0}^{Y-1}\left(\left(g_{X-d_x,l(t)} - g_{X,l(t)}\right)^2 - \left(g_{0,l(t)} - g_{d_x,l(t)}\right)^2\right) \quad (2)$$

Where $m(t)$ is the mean value at the time t, $c(t)$ is the contrast value at the time t, the displacement vector is $(d_x,0)$. X,Y is the width and height of gliding window.And $g_{x,y}$ is the grey value of the pixel whose coordinate in the gliding window is (x,y).

Similarly, we can get also formulation for the vertical direction.

mean: $$m(t+1) = m(t) + \sum_{k=0}^{X-d_x-1}\left(g_{k,Y} - g_{k,0}\right) \quad (3)$$

contrast:

$$c(t+1) = c(t) + \sum_{k=0}^{X-d_x-1}\left(\left(g_{k,Y}(t) - g_{k+d_x,Y}(t)\right)^2 - \left(g_{k,0}(t) - g_{k+d_x,0}(t)\right)^2\right) \quad (4)$$

Therefore to the whole image, an interesting thing happened. If the texture features of lower-left window is calculated, then texture features of other windows can be computed with equations (1), (2), (3), (4). It will greatly reduce the time cost of computations of texture features.

4. GRAPHICS HARDWARE IMPLEMENTATION

Usually, different displacements are used to avoid the problem concerning improper spatial resolution [10]. In this paper, two displacement vectors are used, (2,0) and (4,0).So there are 4 features in each gliding window, they are mean and contrast corresponding to the displacements. They fit well with 4 components of a pixel color value. There is to say, the features of a gliding window will be stored in the RGBA components of its center pixel. In order to get the exact pixel value of the image, we use an orthographic view (no perspective adjustments).The image is converted to texture. Our algorithms are implemented in the fragment shader (NV_fragment_program).

4.1. Increment of the features

Suppose the gliding window size is 15X5.The displacement vectors are (2,0) and (4,0).The grey values for one row pixel are *p0,p1,p2 … p14*.Because the next horizontal window is involved in, so the first pixel value at the same row of next horizontal window is labeled as *p15*.

1) . 1-pass algorithm

According to equations (1) and (2), there are 2 steps to compute the mean and contrast value of a window: in each row, we calculate *p13-p0/p11-p0* for mean value and (*p15-p13)*^{2}*-(p2-p0)*^{2}*/(p15-p11)*^{2}*-(p4-p0)*^{2} for the contrast value;then sum up the values of 5 rows respectively.With fragment shader ,the program is easy to be realized.Fig. 2. is a sample flow chart.

But there are too many instructions in 1-pass algorithm. Totally there are 30 texture fetches, 30 SUBs,18 ADDs and MADs instructions. It has over 100 instructions. The speed is very low on our graphics.

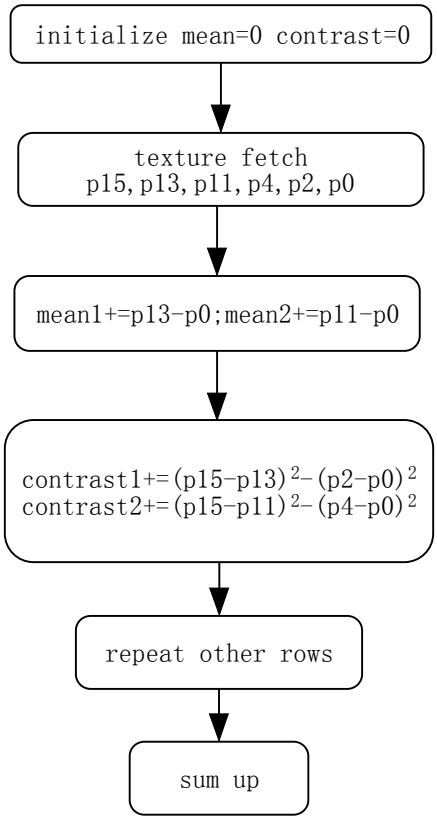


Fig. 2. 1-pass algorithm

After studying the algorithm carefully, we find there are many repetitions in 1-pass algorithm. For example, with two vertical neighboring windows, there are 4 rows that are computed twice. So how to avoid the repetitions is the key point.

2). Multi-pass algorithm

Suppose we apply one texture operation on all texels of image, then the effect is the same as operating on each pixel. For example, in the 1-pass algorithm the values of *p13-p0* of all 5 rows must be calculated, so we can use a SUB instruction in fragment shader program with 2 texture fetches. It will function samely as 1-pass algorithm. But because it avoids the repetitions the efficiency is improved. Then we can sum up the values of 5 rows in another fragment shader program. That is to say, we can decompose the 1-pass algorithm into multi-pass algorithm.

Sub and square operations

For each pixel, it is treated as *p15*. At the same time, four texture units are fetched,*p15,p13,p11,p0*.Then we calculate the values of *p13-p0, p11-p0, (p15-p13)*^{2}*, (p15-p11)*^{2} .The four results are stored as *RGBA* components in each pixel. The values of *(p2-p0)*^{2} and *(p4-p0)*^{2} are automatically included in the above computation.See Fig. 3..

Sub operations

In this program, the sub operations between *(p15-p13)*^{2} and *(p2-p0)*^{2} and between *(p15-p11)*^{2} and *(p4-p0)*^{2} are computed. Where *(p2-p0)*^{2} is stored in the *R* component of *p2, (p4-p0)*^{2}

is stored in the *G* component of *p4*.There are 3 TEXs，2 ADDs instructions.

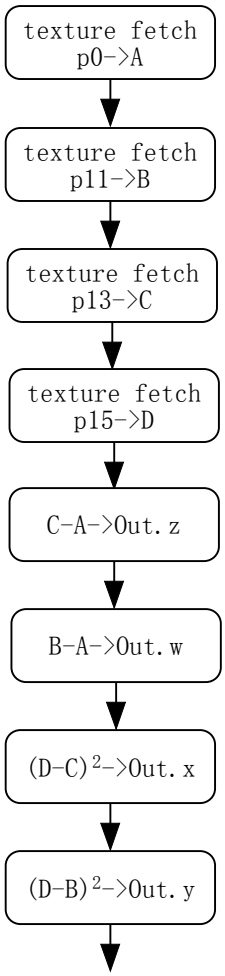


Fig. 3. Sub and square operations

"Out" is the pixel color, *x*,*y*,*z*,*w* are *R G B A* components of it.There are 4 TEXs, 4 SUBs,2 ADDs,2 MULs instructions.

Sum operations

Finally we sum up the value of 5 rows.The pixel that enters this program is treated as the center pixel of the gliding window.That is to say, if its coordinate is *(x,y)*,then we must sum up the values of ($x\pm7$,$y\pm1$), ($x\pm7$,$y\pm2$), ($x\pm7$,y). There are 5 TEXs，6 ADDs instructions. We have completed the computation of increment of features. Compared with the 1-pass algorithm, the total instructions of multi-pass algorithm are 28 instructions, which are almost 1/4 of the original algorithm.

4.2. Iteration of features

Since we have computed the increment of features for each gliding window in every row, the only thing need to do is to calculate the features of first window in each row. Then with iterations we can get all values. The feature values of lower–left window are caculated with CPU. For other rows, with equations (3) (4) ,the increment of feature values of first column windows can be calculate similarily as above. But because the number of first windows are not quite large (512), so we can compute them with CPU.In fact, the time cost of this process is about 1ms.

How to calculate the feature values of each gliding window is a typical Summed Area Tables problem[11].For pixel values at columns greater than 1, they depends on the previous column values. Unfortunately this process can not be realized in the texture operation quickly. So we can only do it with CPU. Firstly, we read buffer into the main memory (GPU->CPU), then with iteratation we can get feature values of each window, then generate new texture with feature values (CPU->GPU).

5. RESULTS

We tested our algorithm with a common PC that has a P4 920 2.8Ghz CPU, 1G ram and a nVidia Geforce 7900GT. The image size is 512X512. The displacement vectors are (2,0) and (4,0).The gliding window size is 15X5. The results are shown in Table 1..

Table 1. Time cost of CPU and GPU

step		Time(ms)
Computation of first window in each row		0.54
increment	Computation in each row	1.28
	Sum of 5 rows	1.01
Iteration of features	Read buffer	4.25
	iteration	14.8
	texture Generation	8.1
total		29.98

From the data in Table 1., one can see the time mainly spend in texture readback and generation, which occupies 12.35ms. The other expensive step is iteration. These two steps are used to solve the Iteration of features. This is because the SAT problem can not be realized in the texture operation quickly. In despite of this, the total time is less than 33.3ms. That is to say, if a image process system consists of the GPU, the speed of it can achevive 30fps. Otherwise, the SLI technology makes it possible for two GPUs works together to get faster speed. While CPU usually has many tasks, it serves as the main processor in PC and can not be used as computation independently. So GPU can be regarded as a new hardware in real-time systems, such as defect detection in automation.

We must point out that our hardware are not expensive. An high-end hardware in consumer market such as Geforce 7950GX2 will be much more faster. The time cost of transferring data between main memory and GPU is the bottleneck. If GPU can complete the iterations, then we will get faster speed. So we believe GPU will have a bright future in real-time systems.

6. CONCLUSION

7. REFERENCES

[1] J. Iivarinen, K. Heikkinen, et al, A Defect Detection Scheme for Web Surface Inspection. International Journal of Pattern Recognition and Artificial Intelligence, September 2000. 14（6）, 735-755.
[2] C C Gotlieb, HeKreyszig, textures descriptors based on co-occurrence matrices, computer vision, graphics, and image processing, 1990, 51: 70-86
[3] E. S. Larsen, D. McAllister, Fast matrix multiplies using graphics hardware, The International Conference for High Performance Computing and Communications, 2001.
[4] M. Hopf, T. Ertl., Accelerating 3d convolution using graphics hardware, In IEEE Visualization ’99, 1999, 471–474.
[5] M. Hopf, T. Ertl, Accelerating morphological analysis with graphics hardware, In Workshop on Vision, Modelling, and Visualization VMV ’00, 2000, 337–345.
[6] T. Purcell, I. Buck, W. Mark, et al, Ray tracing on programmable graphics hardware, In Proceeding of SIGGRAPH 2002, ACM, 2002.
[7] J. Lengyel, M. Reichert et al, Real-time robot motion planning using rasterizing computer graphics hardware, In Proceedings of SIGGRAPH 1990,1990, 327–335.
[8] Jeffrey Bolz, Ian Farmer et al, Sparse matrix solvers on the GPU: Conjugate gradients and multigrid, In Jessica Hodgins, editor, Siggraph 2003, Computer Graphics Proceedings, 2003.
[9] K. Heikkinen, P. Vuorimaa, Computation of two texture features in hardware, Proc. 10th Int. Conf. Image Analysis and Processing, Venice, Italy, September, 1999,27-29.
[10] A.Visa, Identification of stochastic textures with multiresolution features and self organizing maps, Proc. 10th Int. Conf. Pattern Recognition, 1990,(1): 518-522.
[11] Crow, Frank, Summed-Area Tables for Texture Mapping, Proceedings of SIGGRAPH `84, In Computer Graphics 18, 3 July 1984, 207-212.

All Phase Cosine Bi-Orthogonal Transform and Its Application in Image Compression

ZhengXin Hou ZhiYun Gao AiPing Yang
School of Electronic and Information Engineering, Tianjin University, Tianjin China
Email: gaozhiyun123@hotmail.com

ABSTRACT

This paper introduces the concept of all phase cosine bi-orthogonal transform (APCBOT) and an image compression algorithm based on APCBOT. The proposed APCBOT is derived from convolution algorithm of Discrete Cosine Sequence Filter（DCSF）, which is simpler than conventional DCT and free of waveform distortion. In the proposed image-compression scheme, as an alternative of the DCT in conventional JPEG, APCBOT can appropriately attenuate the high frequency components while transforming the original images from time domain into frequency domain, as a result the quantization steps can be very simple. The experimental results show that the APCBOT based compression is superior to DCT based compression in the following aspects：simpler quantization, higher compassion rate and reconstruction image with better quality.

Keyword: image compression, bi-orthogonal transform, APCBOT, quantization, compression ratio.

1. INTRODUCTION

Discrete Cosine Transform (DCT) has been widely used in several signal processing fields such as data compression, feature selection etc., but digital filtering based on DCT is not recommended generally. The reasons are as follows: nonlinear phase characteristic, blocking problems and high complexity. In order to apply excellent characteristics of DCT to digital filtering effectively, a cosine convolution algorithm based on DCT/IDCT, which is designed to deracinate the defects described above, is used to design DCSF [1, 2, 3, 4].

In the proposed image compression algorithm, DCSF is used as the transform matrix: APCBOT. The quantization is simplified greatly and the reconstructed image is as good as or better than that in conventional algorithm, because of the special characteristics of DCSF. In short, under most circumstances the APCBOT outperforms the DCT-based schemes.

2. APCBOT

APCBOT is derived from the convolution algorithm of DCSF, which speeds up conventional sequency filtering and avoids waveform distortion.

2.1 The designing of DCSF

The cosine sequency filter based on DCT/IDCT is signaled as follows: $[Y_L]=C^T\{F\bullet CZ_L\}$ (1)

where $[Z_L]$ is the L_{th} segment of data vector, with the length of N, which is derived form the blocked time domain discrete signal; $[C]$and $[C^T]$ are the $N\times N$ transform matrixes for DCT and IDCT respectively; $[F]$ is the sequence response vector with the length of N; $\bullet$ is the symbol of inner product; $[Y_L]$ is the output vector of filter and it is the element of the overall output signal of filter.

The i_{th} component of $[Y_L]$ is:

$$Y_L(i)=\sum_{k=0}^{N-1}C^T(i,k)\left\{F(k)\sum_{j=0}^{N-1}C(k,j)Z_L(j)\right\}$$
$$=\sum_{j=0}^{N-1}Z_L(j)\sum_{k=0}^{N-1}C^T(i,k)C(k,j)F(k) \quad (2)$$

Let $H_i(j)=\sum_{k=0}^{N-1}C^T(i,k)C(k,j)F(k)$ i, j = 0,1,⋯, N -1 (3)

Then $Y_L(i)=\sum_{j=0}^{N-1}H_i(j)Z_L(j)=\langle H_i,Z_L\rangle$ $i=0,1,\cdots,N-1$ (4)

There are N different data blocks for different beginnings of the block, and they all contain certain point $z(n)$ of the input signal. In order to utilize Eq. (4) later, it is necessary to denote the data block with partial overlapping in the form of $\{Z_L, L=0, 1,\ldots, N-1\}$, that is:

$$\left.\begin{aligned}&[Z_0]^T=[Z_0(0),Z_0(1),\cdots,Z_0(n-1)]=[z(n-N+1),z(n-N+2),\cdots\cdots,z(n)]\\&[Z_1]^T=[Z_1(0),Z_1(1),\cdots,Z_1(n-1)]=[z(n-N+2),z(n-N+3),\cdots,z(n),z(n+1)]\\&\cdots\cdots\\&[Z_{N-1}]^T=[Z_{N-1}(0),Z_{N-1}(1),\cdots,Z_{N-1}(n-1)]=[z(n),z(n+1),\cdots\cdots,z(n+N-1)]\end{aligned}\right\} \quad (5)$$

By the above data blocks and Eq.(4), there are N different filter results: $Y_0(N-1)$, $Y_1(N-2)$,…, $Y_{N-1}(0)$ at the same point, due to the different beginning points of the block. In order to eliminate the effect of different beginning points of the block in filtering, the average of N filer data is chosen as the filtering response $y(n)$,

That is $y(n)=\frac{1}{N}\sum_{i=0}^{N-1}Y_i(N-1-i)$

By Eq.(4) andEq.(5), we have:

$$\begin{aligned}y(n)&=\frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}H_{N-1-i}(j)Z_i(j)\\&=\frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}H_{N-1-i}(j)z(n-N+1+i+j)\\&=\sum_{k=-(N-1)}^{N-1}Q(k)z(n-k)=Q(n)*z(n)\end{aligned} \quad (6)$$

Where $\left.\begin{aligned}Q(n)&=\frac{1}{N}\sum_{m=n}^{N-1}H_m(m-n)\\Q(-n)&=\frac{1}{N}\sum_{m=0}^{N-1-n}H_m(m+n)\end{aligned}\right\}$ $0\le n\le N-1$

Eq.(6) is the convolution form of DCSF, which means that the output signal $\{y(n)\}$ is the convolution of the time-domain discrete signal $\{z(n)\}$ and DCSF $\{Q(n),-N+1\le n\le N-1\}$. DCSF is FIR filter with the length of $2N-1$. The DCSF not only can be implemented easily, but also eliminates the block sign of conventional sequence filter, which improves filter greatly.

Concerning DCT,

$$C^T(i,j)=\begin{cases}\dfrac{1}{\sqrt{N}} & j=0, 0\le i\le N-1\\ \sqrt{\dfrac{2}{N}}\cos\dfrac{j(2i+1)\pi}{2N} & 1\le j\le N-1, 0\le i\le N-1\end{cases} \quad (7)$$

By Eq. (3), we have

$$H_i(j)=\frac{1}{N}\left[F(0)+\sum_{k=1}^{N-1}2\cdot\cos\frac{k(2i+1)\pi}{2N}\cdot\cos\frac{k(2j+1)\pi}{2N}\cdot F(k)\right] \quad (8)$$

By $H_j(j)=H_i(j)$ and Eq (7),

We have $Q(n)=Q(-n)$ $0\le k\le N-1$ (9)

Therefore, the DCT filter *{Q (n)}* is linear in phase. Due to the symmetry characteristic, the whole DCSF is known only if the $Q_{1/2}$ is known, where $[Q]_{1/2}=[Q(0),Q(1),\ldots Q(N-1)]^T$.

The definition of matrix R and matrix S are as follows:

$$R(m,n)=\begin{cases}\dfrac{1}{2} & n=0,\quad 0\le m\le N\\ \cos\dfrac{mn\pi}{N} & 1\le n\le N-1,\quad 0\le m\le N\end{cases}$$

$$[S]=\begin{bmatrix}N & 2 & 0 & 2 & 0 & \ldots & 0 & 2 & 0\\ 0 & N-1 & 2 & 0 & 2 & \ldots & 2 & 0 & 1\\ 0 & 0 & N-2 & 2 & 0 & \ldots & 0 & 2 & 0\\ 0 & 0 & 0 & N-3 & 2 & \ldots & 2 & 0 & 1\\ \ldots & \ldots & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & \ldots & 2 & 2 & 0\\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 1 & 1\end{bmatrix}$$

By Eq.(7) and Eq.(8), we have

$$[Q]_{1/2}=\frac{1}{N^2}[S][R][F]=\frac{1}{N^2}[V][F] \quad (10)$$

Where matrix $V=SR$ is $V(m,n)=$

$$V(m,n)=\begin{cases}N-m\\ (N-m)\cos\dfrac{mn\pi}{N}-\csc\dfrac{n\pi}{N}\sin\dfrac{mn\pi}{N}\end{cases} \quad (11)$$

$$\begin{cases}0\le m\le N-1, n=0\\ 0\le m\le N-1, 1\le n\le N-1\end{cases}$$

Hence, Eq. (9), (10) and (11) make up of the 1-D DCSF.

2.2 APCBOT

According to the Eq. (11), it is plain that the row vector is of sequency characteristic, which means that the sequency will increase as the row index increasing and the altitude change will decrease as the sequency increasing. Therefore, we define the 1-D APCBOT: $F=[A][f]$ and the inverse APCBOT: $f=[A^{-1}][F]$, where $A=V/N^2$. Further, the row vector of A is resolutive base vector and the column vector of $A^{-1}$is resultant base vector. They are both bi-orthogonal and of sequency characteristic and dualization characteristic. So it is accepted to apply the resultant base vector to reslove the original image and the resolutive base vector to synthesize the reconstructed image. In a word, APCBOT is make up of A and A^{-1} [5, 6].

In order to simplify the quantization matrix, DCT is replaced by APCBOT in the proposed algorithm, which means that the transform matrix in the new JPEG algorithm is $A=V/N^2$.When N=8,

$$A(m,n)=\begin{cases}\dfrac{8-m}{64} & 0\le m\le 7, n=0\\ (\dfrac{8-m}{64})\cos\dfrac{mn\pi}{8}-\csc\dfrac{n\pi}{8}\sin\dfrac{mn\pi}{8} & 0\le m\le 7, 1\le n\le 7\end{cases}$$

即,$A=$

$$\begin{bmatrix}0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250\\ 0.1094 & 0.0854 & 0.0617 & 0.0262 & -0.0156 & -0.0575 & -0.0930 & -0.1167\\ 0.0938 & 0.0374 & -0.0221 & -0.0783 & -0.0938 & -0.0543 & 0.0221 & 0.0952\\ 0.0781 & -0.0078 & -0.0709 & -0.0657 & 0.0156 & 0.0787 & 0.0396 & -0.0676\\ 0.0625 & -0.0408 & -0.0625 & 0.0169 & 0.0625 & -0.0169 & -0.0625 & 0.0408\\ 0.0469 & -0.0557 & -0.0175 & 0.0498 & -0.0156 & -0.0368 & 0.0488 & -0.0198\\ 0.0313 & -0.0510 & 0.0221 & 0.0101 & -0.0313 & 0.0341 & -0.0221 & 0.0068\\ 0.0156 & -0.0301 & 0.0267 & -0.0216 & 0.0156 & -0.0096 & 0.0046 & -0.0012\end{bmatrix}$$

When comes to 2-D image data, the forward transform of APCBOT: $F=[A][f][A^T]$, and the inverse transform is: $f=[A^{-1}][F][(A^{-1})^T]$.

3. IMAGE COMPRESSION WITH APCBOT

3.1 The JPEG algorithm based on APCBOT

The JPEG algorithm based on APCBOT is the same as that based on DCT, expect for the transform matrix and the way of quantizing [1]. The process is as follows: transforming the image form time domain into frequency domain with APCBOT; quantizing the APCBOTed coefficients with uniform quantization table or without quantization table; DPCM coding of the DC coefficient difference and Huffman coding of the AC coefficient zigzag-scanned.

With APCBOT the high frequency components of the original image will be attenuated according to their frequency values. Take the 8×8 coefficient block for instance. The coefficients in each row will be attenuated about one to eight times from left to right respectively, and likewise the coefficients in each column will also be attenuated about one to eight times from up to down. Therefore, the coefficients in the block will be attenuated increasingly from the top left one to the bottom right one. In the DCT-based algorithm, larger quantization steps are applied to high frequency components to lower code ratio. Since APCBOT can attenuate high frequency components, uniform quantization table can be used to simplify the algorithm, which makes the adjustment of code ratio undemanding.

Table 1. SimulationResults under Different Circumstances

	Compression Ratio			Bit Rate			PSNR		
	Ⅰ	Ⅱ	Ⅲ	Ⅰ	Ⅱ	Ⅲ	Ⅰ	Ⅱ	Ⅲ
cornfield	7.3624	12.0332	7.5240	1.0866	0.6648	1.0633	35.1126	34.3896	35.3366
boats	8.6888	15.3645	8.4587	0.9207	0.5207	0.9458	34.7255	34.2039	35.0274
peppers	10.6231	18.8312	10.9959	0.7531	04248	0.7275	34.7759	34.3036	34.8065
girl4	21.4560	33.8551	24.9714	0.3729	0.2202	0.3204	40.4828	40.9968	40.7900
model	17.1584	29.5619	19.3779	0.4662	0.2706	0.4128	40.6479	40.0241	40.8015
mountain	3.0223	4.9347	2.6479	2.6470	1.6212	3.0213	25.1261	24.5904	25.4885
miramar	3.7693	6.8082	3.2205	2.1224	1.1750	2.4841	27.0520	26.7873	27.4878
lena	10.6024	19.3582	10.8906	0.7545	0.4133	0.7346	35.8050	35.1557	35.8167

Note: Ⅰ corresponds to JPEG algorithm based on DCT; Ⅱ corresponds to APCBOT with $A=V/N^2$ and without quantizaion table; Ⅲ corresponds to APCBOT with $A=V/N$ and uniform quantizaion(Q=52) table;

Fig. 1. Lena

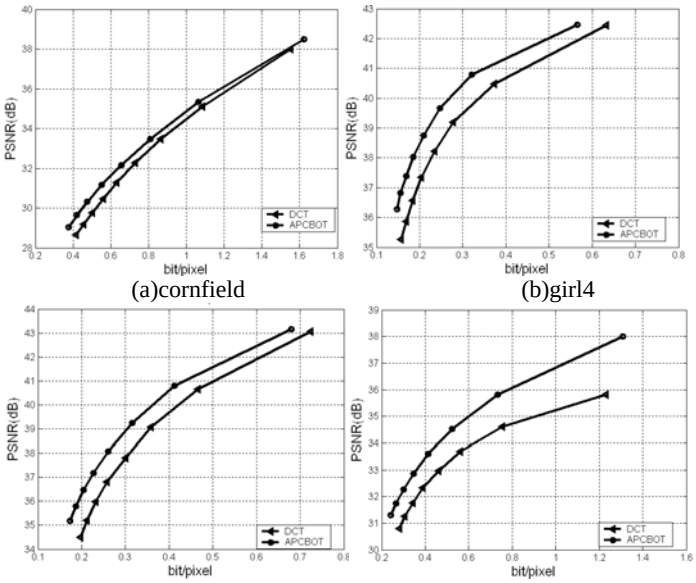


Fig. 2. Rate-distortion graph

3.2 JPEG algorithm based on APCBOT without quantization talbe

Experiments done in Matlab with 8 grayscale images(512×512×8), show that when matrix A is used as transform matrix the compression ratio and the quality of reconstructed image are good enough for common applications even without quantization . The detailed data is shown in Table 1.

For the randomly selected eight images, the compression ratio is almost doubled in the JPEG algorithm based on APCBOT compared with the DCT-based one, PSNRs for these two circumstances do not differ much and the perceived qualities of reconstructed images in different ways are almost the same, as is shown in Fig. 1. Therefore, with this algorithm without quantization, image processing will be sped up and it makes transmit image with narrower bandwidth possible.

3.3 The JPEG algorithm based on revised APCBOT with uniform quantization table

The above JPEG algorithm based on APCBOT without quantization table also means uniform quantization with the quantization step of '1'. In order to obtain reconstructed images with higher quality, the quantization step should be less than '1' theoretically, which is not allowed. Hence, we have to adjust the transform matrix A. For example, let $A=V/N$. In this way, the transform matrix is obtained by replacing summing with averaging in the all phase filtering.

According to the data in table 1, the reconstructed images obtained from the algorithm based on revised APCBOT with transform matrix $A= V/N$ and uniform quantization table (Q=52) is better than that from DCT-based algorithm. Hence, the new JPEG algorithm outperforms the DCT-based one both in compression ratio and in PSNR. In addition, storage space is reduced because of the uniform quantization step.

Reconstructed images of 'lena' in different transform ways are shown in Fig.1:(a) is the original image 'lena';(b),(c),(d) are the reconstructed images by DCT, by APCBOT without quantization and by revised APCBOT with uniform quantization (Q=52) respectively. It is clear that the reconstructed image by APCBOT is as good as or better than that by DCT.

In order to compare the property of JPEG algorithms based on DCT and APCBOT, we encode and decode images: 'cornfield', 'girl4', 'model' and 'lena' in two ways respectively with different compression ratios. The result is shown as the rate-distortion graphs, in Fig. 2, where the line with dots denotes revised APCBOT with uniform quantization table and the line with triangle denotes DCT. It is clear that the quality of reconstructed images by APCBOT is higher than by DCT under the same bit rate and the compression ratio with APCBOT is lower than with DCT under the same PSNR. In summary, compared with the DCT-based algorithm, the proposed algorithm is much better is two ways: better coding property and simpler quantization.

4. CONCLUSION

A new JPEG algorithm based on APCBOT is presented in this paper. Experiments show that the quantizaion is simple and the coding property is good in the APCBOT-based JPEG algorithm. When the transmission bandwidth is limited and the quality of reconstructed image is not very important for the receiver, the JPEG algorithm based on APCBOT without quantization table is advisable.

5. REFERENCES

[1] ISO/IEC 10918-1|ITU-T Rec.T.81 , "Digital Compression and Coding of Continuous-Tone Still Images"
[2] N. Ahmed, T. Natarajan, et al, "Discrete Cosine Transform" , IEEE Trans. Compute vol.23(1):90-93
[3] N.S. Jayant, P. Noll, "Digital Coding of Waveforms". Englewood Cliffs, NJ: Prentice-Hall, 1984.
[4] Byeong Gi Lee, "A new algorithm to compute the discrete cosine transforms." IEEE Trans. ASSP, 1984, 32(6): 1243 -1245.
[5] N. Abmed, K. R. Rao, "Orthogonal transform for digital signal processing". Berlin; Springer, 1975.
[6] Z.X. Hou, Z.H. Wang, et al, "Design and implementation of all phase DFT digital filter". Acta Electronic Sinica, 2003, 31(4): 539-543.

Design and Realization of Visual Training Simulation system

Yuan Shengzhi, Xie Xiaofang
Department of Science and Technology of Weapons, Naval Aeronautical Engineering Institute
Yantai, Shandong Province, China
Email: yuanshengzhi_hy@sina.com
Bai Xiaoming
The 92840th troop of the PLA;
Qingdao, Shandong Province, China
Email: yjl7855@sina.com

ABSTRACT

Proposed the application of Distributed Virtual Reality in the field of military aviation. Introduced the structure of the visual training simulation system for airborne assisting policy based on the High Level Architecture (HLA). Also introduced how to build the 3D simulation models by using Creator, how to use Vega into the vision processing, how to integrate the whole system with VC++. The Training System, which was developed, may be in favor of improving the battle effectiveness of our aviators in flight, which is much important for mastery of sky.

Keywords: HLA, Vega, Creator, Training Simulation System.

1. INTRODUCTION

With the development of science and technology, the level of aerial weapons has been improved. The technological content of aerial weapons is much higher than ever, the sys-tem of aerial weapons becomes increasingly complex, and the cost of aerial weapons rises quickly. How to improve the tactical ability of our aerial army based on the good status of the equipments is a momentous problem, which needs be solved badly. In fact, it is not a reality to use the true equipments for training. Therefore, it is much necessary for us to do research on training simulation system based on distributed VR technique for airborne assisting policy, which is much necessary to improve the training level of our aerial army [1].

The technique of Virtual Reality (VR), which is based on Computer Graphics, Multimedia, AI (artificial intelligence), Human-Machine interface, Sensor, parallel real-time computing and the action of person[2], is one of the highest active fields in these years. In their book “Virtual Reality Technology” Burdea G. and Coiffet P. have put forward the concept of “Virtual triangle”, which had explained compactly the basic characters of Virtual Reality: Immersion-Interaction-Imagination (in Fig.1.). It has given the users more vivid experiences; it has given us a much easier interface; it has created a new, strange, multi-perceptive simulation world, in which we can be deeply immersed. It has also created a new way to do research on simulation and training of weapons.

We propose a visual training simulation system for airborne assisting policy based on Distributed VR technique. In order to build the architecture of the system, we plan to use the High Level Architecture (HLA) for the interactive simulation, to use Creator, Vega for Image and vision processing. The paper is organized in the following way: Section 2 presents the design of the training simulation system based on HLA. In Section 3 we propose the application of Vega, Creator and VC++ in the design of visual simulation. In Section 4 we summarized issues mentioned in this paper and describe plans for further work.

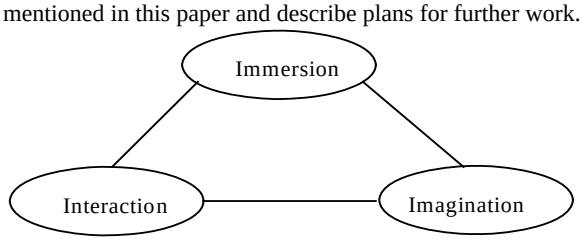


Fig. 1 Basic Characters Triangle of Virtual Reality

2. DESIGH OF THE VISUAL TRAINING SIMULATION SYSTEM BASED ON HLA

Characters of the Training Simulation System

The Training Simulation System for airborne assisting policy is used to do training on the aviators for tactics. This system can give us a virtual oppositional environment, which the aviators are deeply dropped into. At the same time, the aviators follow the result of the airborne assisting policy system, and do the relational actions, such as elusion, fighting, and electro-disturbance in order to enhance the battle effectiveness.

Based on the demands about the abilities of the Training Simulation System, the system we designed has characters as following:

1) This Training Simulation System is a typical distributed simulation system for training, also a person-in-loop, interactive, net-counterwork simulation.

2) This Training Simulation System is a typical system of Virtual Reality (VR). Thus, it has characters of the system of VR.

Structure of the Training Simulation System based on HLA

With the development of simulation mode, large-scale simulation is more complex than ever. In order to fulfill the needs of Distributed Interactive Simulation, the Distributed Interactive Simulation protocol (DIS) came into being twenty years ago. Though the DIS has much merit, like its predecessor SIMNET, which is designed for a very specific application domain: military simulations, it can’t fulfill the needs of different simulation model reusing and inter-operation. So the DMSO issued the High Level

Architecture [3] (HLA) of the system modeling and simulation in 1996.8, which has been the standard of the IEEE. The High Level Architecture (HLA) is composed of three parts: HLA Rules, Interface Specification and Object Model Template of the HLA [4, 5].

According to the particularity of the Training Simulation System, especially the diversity of the counterwork, we propose the HLA to build its structure. Based on the Federation Development and Execute Process Model (FEDEP) of the HLA [4, 5], we defined the Training Simulation System as a Federation, which is composed of five federation objects: the object of the aviator system, the object of the airborne assisting policy system, the object of the battlefield environment, the object of the counterwork target, the object of the simulation controller. The objects of the Federation and Run-Time Infrastructure form a distributed interactive simulation system----the Training Simulation System. The structure of the Training Simulation System is showed as Fig. 2..

In Fig. 2., the federation object of the simulation controller is composed of four parts: Visual System of the Training Simulation, the controller of the Training Simulation, Recording system, and interface of RTI. The controller of the Training Simulation takes charge of the process of the simulation, the initialization of the simulation, which is about the targets of the training and the virtual environment of the simulation. The Visual System of the Training Simulation takes charges of the present of the training simulation. It offers the scene of the simulation immediately through the projectors and the spar screen. Recording system can record all the scenes during the course of the training simulation, which can give us the chance to evaluate the effect of the training.

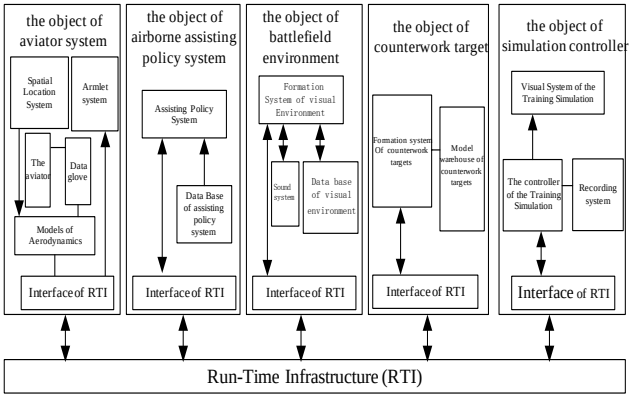


Fig. 2. The structure chart of the training simulation system

The federation object of counterwork targets is about three parts: Formation system of counterwork targets, Model warehouse of counterwork targets and interface of RTI. According to the targets and the virtual environment of the training simulation, which was initialized by the object of the simulation controller, the object of counterwork targets chooses the relative model from the Model warehouse of counterwork targets, and forms the real visual targets through Formation system of counterwork targets.

The federation object of battlefield environment can offer a real virtual environment the aviator deeply immersed in. It is composed of four parts: Formation system of visual environment, Database of visual environment, sound system and interface of RTI. According to the targets and the process of the training simulation, which the object of the simulation controller initialized, the object of battlefield environment choose the different models form Database of visual environment, and forms the real visual environment. It also provides us with the real sound environment through sound system.

The federation object of airborne assisting policy system is composed of three parts: Assisting policy system, Database of assisting policy system and interface of RTI. Based on the real visual environment of the battlefield, it chooses different data from the Database of assisting policy system; does decision-making and gives the object of aviator system relative cues.

In the federation object of aviator system, the aviator is seen as a part, a part of person-in-loop simulation. The object is composed of six parts: the aviator---the main body of the Training Simulation System, Data glove, Spatial Location System, Armlet System, Models of aerodynamics and interface of RTI. The object can get the visual environment from the object of battlefield environment; receive the cues form the object of airborne assisting policy system; get the input data through Data Glove. Based on all the information, the object calculates the location and the pose changes of the airplane through Models of aerodynamics. At last, we get the changes, which is interactive and drive the process of the Training simulation system.

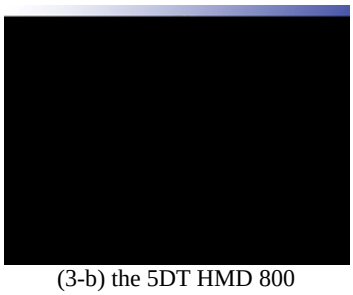
Hardware and software platform of the Training Simulation System

Hardware Platform of the Training Simulation System: In the process of the design, we chose the Hardware platform as following:

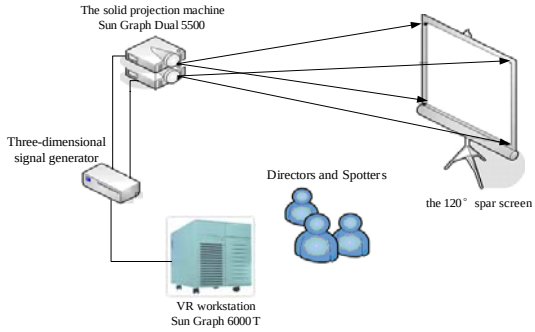
1) 2 sets Sun Graph 6000 series VR workstation--Sun Graph 6000T (CPU: 4500MHz, Cache built in CPU: 2MB,Mian Memory: 2GB, System Bus: 533MHz.Hard Disk: 400G, 1000M Ethernet supported.);
2) 6 sets slap-up computer---Dell OptiPlex GX520 (CPU: 3000MHz, Main Memory: 1GB, System Bus: 533MHz.Hard Disk: 120G);
3) The solid projector---Sun Graph Dual 5500(4 sets), the 120° spar screen;
4) The 5DT Data Glove Ultra[6], the 5DT HMD 800;
5) The 100M Ethernet.

The main equipments of the Training Simulation System are shown as following Fig. 3.. The 5DT Data Glove Ultra is shown as Fig. 3. (3-a); Armlet system --the Sim Eye XL100A (HMD) is shown as Fig. 3. (3-b). In the Fig. 3. (3-c), VR workstation--Sun Graph 6000T, The solid projector---Sun Graph Dual 5500, the 120° spar screen are used to form the Visual Simulation System of the Federation object-- the simulation controller.

(3-a) the 5DT Data Glove Ultra

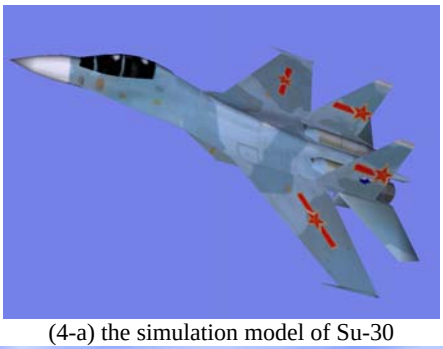


(3-b) the 5DT HMD 800



(3-c) Structure of Visual Simulation System

Fig. 3. The main equipments of the Training Simulation System



(4-a) the simulation model of Su-30

(4-b) the simulation model of the artillery

Fig. 4. Simulation models developed by Creator

Software platform of the Training Simulation System: Based on the Operating System Windows NT WorkStation 4.0, the software platform: Visual C++ 6.0 and Vega 3.70, was chosen. In the course of software designing, the MFC method of Visual C++ was used for interfaces designing and functions integrating. The 3D simulation models were developed by Creator 3.0. Expect driving the visual Simulations, Vega3.70 has other functions: checking out the collisions, dealing with LOD, loading the special effects such as missile trail, explosion, waves and environment effects, taking charge of the Large Area Database Management (LADBM).

3. KEY POINTS IN THE COURSE OF SYSTEM DESIGNING

Technique of Modeling by Creator

MultiGen Creator is a software package designed specifically for the creation of real-time 3D models for visual simulation. Creator makes it easy to import, structure, edit, proto-type, and optimize Model databases for use in both large-scale visual simulations [6]. With this software we can easily export *.flt files for further application. The data format of OpenFlight has been the common standard in the VR industries. This kind of files can be imported in the software Vega for further real-time interactive application.

In the course of system designing, we use Creator to build models database of anti target. The models database of anti-target has include the models of battle plane, artillery and aerial defense missile. The two simulation models build by Creator are shown as following Fig. 4..

In the Fig. 4., the simulation model of Su-30 (form Russia) is shown as Fig. 4. (4-a); the simulation model of the artillery is shown as Fig. 4. (4-b).

Image and vision processing by Vega

After the models building, we can import those models into Vega for further development. Vega is a development environment and software library for the creation of real-time visual and audio simulations [7]. It runs under the Microsoft Windows NT and Windows 2000 operating systems on the Intel (Pentium) workstations. Vega NT is targeted for use with the Microsoft Visual C++ Development Environment. It is composed of the graphical user interface Lynx and Vega library [7].

The position of the observer, the nature and placement of players and objects in the scenes, how to make image and vision processing, the lighting set-up, the specification of the environment and environment effects, and the consideration of the target hardware platform, are a few of the many parameters that must be considered. The Lynx helps us manage details concerning the values of these parameters through a graphically oriented point-and-click interface. The following code fragment has defined an observer and the relationship with other objects and the initial parameters.

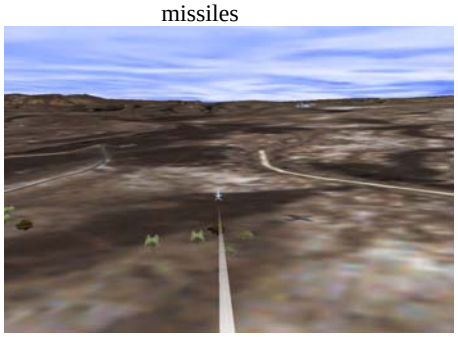
```
Observer Fighter
{
        Channel Default;
        Scene Default;
        Envlocal;    // the parameter of the environment.
        Motion flight;      //motion model UFO.
        Position    0    0    0;
        …
}//the definition of other parameters was overleapt.
```

According to different needs, the difference simulation scenes are designed. Attacking, counter with aerial defense missiles, elusion is the common action and the scenes of the training simulation system. We just chose these typical scenes shown as following.

In Fig. 5., the simulation scene for attacking was shown as Fig. 5. (5-a); the simulation scene for counter with aerial defense missiles was shown as Fig. 5. (5-b); the simulation scene for elusion was shown as Fig. 5. (5-c).

(5-a) the simulation scene for attacking

(5-b) the simulation scene for counter with aerial defense missiles



(5-c) the simulation scene for elusion

Fig. 5. Typical scenes developed by Vega

Software Development based on Simulation scenes

Vega is MultiGen-Paradigm's premier software environment using for the creation, development and deployment of 3d real-time visual and audio simulation, sensor, virtual reality, and general visualization applications.

Vega provides an extensive and mature programming API in C, and a large list of add-on expansion modules.
By combining advanced simulation functionality with easy-to-use tools, Vega provides the basis for the most productive process for building, editing, running and deploying sophisticated applications quickly and easily. Vega is used by leaders in the airline transportation, aircraft manufacturing, space and defense industries utilizing Vega's advanced 3d simulation technology for critical operations such as pilot and gunner training, aircraft design, mission planning rehearsal and missile visualizations.

In order to reuse the simulation scenes developed by Vega, we should get the support of Vega environment. Based on the software platform VC++, for the further application without Vega environment, we should make the Vega library actively. The following code fragment about the console program can be easily understood. A method can be concluded from the code fragment.

```
#include <vg.h>    //the header file of Vega;
main()             // the application of the console method;
{
    vgInitSys();        //the initialization of the Vega Library;
    vgDefineSys("myapp.adf");//the definition of
                        // a application by the *.adf file
    vgConfigSys();//configure the application about Vega
    while(Simulation_Flag)
    {
        vgSyncFrame(); //the application of
                        //functions in the Vega library
        vgFrame();      //the application of functions
                        //in the Vega library
    }
}
```

4. SUMMARY AND FUTURE WORK

Stepping into the 21 century, it is a trend for us to develop new simulation training system to replace the true equipments, to develop the virtual methods for training to replace the actual training methods. With the development of the training simulation system for airborne assisting policy based on Distributed VR technique, we can improve the tactical ability of our aviators without the true equipments. It helps improve the battle effectiveness of our aviators in flight, which is much important for mastery of sky.

In the process of developing the training simulation system, we have use Creator to build the 3D simulation models, use Vega to develop the Visual simulation scenes, and use VC++ to integrate the whole system. It is a common method in the VR simulation applications to use Creator, Vega and VC++, which can give you some reference. In the theory researching, How to apply HLA into the largest scale simulation faultlessly is a further work for us.

5. ACKNOWLEDGEMENTS

The paper belongs to the task of the PLA. Thank to my director—Professor Xie xiaofang, who is the famous scholar in the domain of Virtual and Augmented Reality in the PLA. He gave me a chance to do research on the VR technique and its applications. In the process of developing the visual training simulation system, Cao Jian and Liang Jie gave me many helps. So I show thanks to them.

6. REFERENCES

[1] Bemard P.Zeigler, Multifaceted Modeling and Discrete Event Simulation, Academic Press, 1984， 20-30.

[2] M.R.stytz, “Distributed Virtual Environment”, IEEE Computer Graphic and Applications, 1996, 16(3)：19-31.

[3] IEEE P1516.1 Standard for M&S High Level Architecture Federate Simulation Interoperability Workshop, 99S-SIW-116, March, 1999.Interface Specification [S]. Draft 1，DMSO，April 20, 1998.

[4] Kevin Cox. A Framework-based Approach to HLA Federate Development [R]. Simulation Interoperability Workshop, 98F-SIW-036, March 1998. 120.

[5] Y. Zhou, J.W. Dai, Design of Simulation Program based on HLA, Publishing House of Electronics Industry. Beijing, china, 2002. 11-54.

[6] Vimtek Inc.5DT Data Glove Ultra Series User’s Manual [M], Vimtek Inc, 2004.

[7] MultiGen Paradigm Inc.Vega3.70 programmer’s Guide and Lynx help [M], MultiGen Paradigm Inc, 2001.

A Method Of Triangle Mesh Gap Classification And Mending

Feng Li, Zhifeng Tian, Wenbo Xu

School of Information Technology, Southern Yangtze University

Wuxi,Jiangsu, 214122, China

Email: wx_lifeng@hotmail.com

Wen-hao LENG

IT Divison, China Ship Scientific Research Center

Wuxi, Jiangsu, 214082, China

ABSTRACT

As a result of the complexity of the ship structure analysis models, the logical relationship of the model data is frequently lost in the transformation between the design and analysis software. The holes, slots and small curved surfaces caused by the failure of transformation will create mesh gap in the model, causing calculation divergence. Meshing of the ship model by universal finite element mesh (FEM) software shows that the number of gaps is comparatively small in relation to triangle and each gap group consists of only a few nodes. The paper proposes one new method using adjacency matrix to group gaps found in whole FEM analysis model, create mesh gap adjacency matrix by connected weight, class the gaps according to connectivity, and then regroup connected components by articulation nodes. BFS algorithm and marked array are applied to adjacency matrix traversing while mending triangle methods are provided. The results show that it is more effective and better for meshing complicated models.

Key words: FEM, triangle, gap, node, adjacency matrix.

1. INTRODUCTION

Application of FEM in ship structure analysis is developing. 3-D FEM model for large container ship, by adding loads and adjusting inertia, is used to calculate its longitudinal bending distortion, shear stress distribution, and the efficiency of the superstructure to stand against longitudinal strength[1]. Study of the effect of non-symmetric flood and rigidity loss of damaged ship hulls to external loads based on FEM analysis[2]. CAD software, with solid modeling including parametric modeling of feature enhancing and freedom surface modeling as example, is better at the representation of surface morphology than CAE software. It is a practical method to apply model design in CAD software such as CATIA、PRO/E, complete FEM analysis in general FEM software such as NASTRAN、MARC. In the transformation between CAD and CAE software, geometry can deliver easily, the logical relationship of the model data is frequently lost. The holes, slots and small curved surface caused by the failure of transformation will create triangle mesh gap in FEM analysis model. Model mending and finite element meshing is important for FEM numerical simulation. They affect the accuracy of the result. Research of mending the damaged triangle model by eliminating detail character，holes and sewing small surfaces into another surface already have some achievement[3,4].

Study and improve traditional triangular mesh rebuilding method and gap discovering method, A new method that grouping mesh gap using adjacency matrix is provided. Gap will be classified by twice grouping. Triangle mending method is developed according to gap classification, element shape, aspect ration and edge angle.

2. BASIC PRINCIPLE AND DEFINITION

2.1. Triangle mesh principle

Assuming $\{P_i\}_{i=1}^{n}$ is a point set on Euclidean, which meet the Voronoi diagram conditions, $V_j = \mathop{\mathrm{I}}_{1\le j\le n, j\neq i} \left\{P : \|P - P_i\| < \|P - P_j\|\right\}$, all points in $\{P_i\}_{i=1}^{n}$ are not in a straight line, connect P_i, P_j when the two points have common edges, there is Delaunay triangulation as the following two criterions are meet.

1.Empty circum-circle criterion. There exists only one triangulation connection created by the point set V_j. Circum-circle of the triangulation does not contain other vertexes.

2. Max-min angle criterion of triangulation. The sum of min

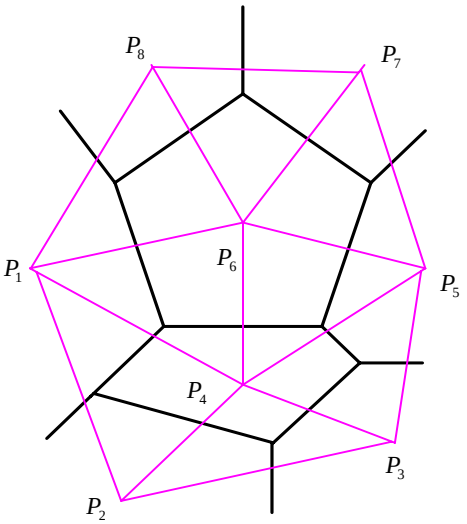


Fig. 1. Voronoi diagram and Delaunay triangle

angle of all triangles that created by point set V_j is maximal.

The two Delaunay criterions reduce thin-layered triangle elements, and add equilateral triangle elements.

2.2. Definition of Gap Related

Def 1. Adjacency matrix A.
Describe the connectivity of the nodes related to the gap in Delaunay triangle mesh.

$$A_{i,j} = \begin{cases} 1 & \mathrm{P_i},P_j \textit{ are only belong to the same triangle} \\ 0 & \mathrm{i}=\mathrm{j} \textit{ or do not meet the condition above} \end{cases}$$

Topology of adjacency matrix is described as undirected connected graph G. and the matrix is a diagonal symmetric matrix. P_i, P_j are interconnected when $A_{i,j}=1$, and an edge e is existing. Nodes connectivity have transitivity, $A_{i,j}=1$, $A_{j,k}=1$,so $A_{i,k}=1$, P_i and P_k are interconnected.

Def 2. Marked array and Degree of Node.
Marked array record the visited information of the nodes,

$$T_i = \begin{cases} 1 & \textit{Node i have already been visited} \\ 0 & \textit{Opposite} \end{cases}$$

Degree of the node equal to the amount of edge that the node connected, $\mathrm{TD_i} = \sum_{j=1}^{n} A_{i,j}$ 。

According adjacency matrix A and degree TD_i, edge e is may estimation, $e = \frac{1}{2}\left\{\sum_{i=1}^{n} TD_i\right\}$

Def 3. Articulation node related to the mesh Gap graph.
In undirected connected graph G, when node P_i and connected edges are deleted, if the graph can be divided into two or more connected components, then P_i is the articulation node. In Fig.2, shadow is the mesh gap, others are triangle elements, P_1 is a articulation node.

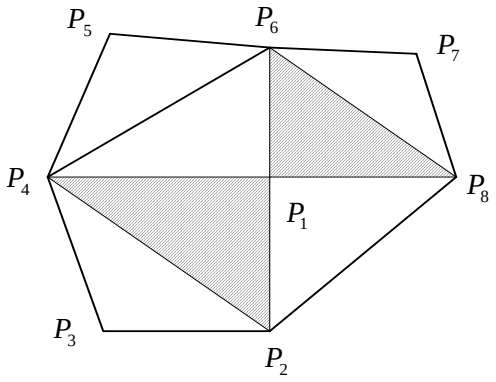


Fig. 2. Articulation node in connected component

3. GAP DISCOVERY

After all the surfaces of a whole ship model are generated into triangle mesh, a triangular mesh model do not exist gap when arbitrary edge belongs to two triangular elements. If one edge belongs to one triangular element only, this edge belongs to a gap polygon border (boundary edge). These border edges connected end to end compose a space polygon, described as gap (or hole), which is the reason of calculation divergence of structural strength calculation of the whole ship model. Research of gap mending provides some algorithms [5,6]. Existing algorithm abstract the gap to space polygon, then fill the polygon with triangular. Rebuilding grid lost its integrity with the original model, and could not meet the analysis needs of FEM software. In general, gap is little in respect of the whole elements，and only few nodes exist in one gap group.

4. GAP CLASSIFICATION

4.1. Build Adjacency Matrix and Select Traversing Algorithm

Searching the ship structure model, find the node set $P\{P_1, P_2, ...P_n\}$ and triangle mesh set $Elm\{Elm_1, Elm_1, ...Elm_m\}$, which are related to the mesh gap. According Def1, A is a diagonal symmetric n×n adjacency matrix. For each node related to the gap N_i, searching in Elm, connectivity weight of P_i and P_j in the gap is $W_{i,j} = \sum_{k=1}^{m}(W_{Edge\,i,j})$, and

$$W_{Edge\,i,j} = \begin{cases} 1 & \mathrm{N_i}\in Elm_k, \mathrm{N_j}\in Elm_k, i\neq j \\ 0 & \mathrm{N_i},\mathrm{N_j}\notin Elm_k \textit{ at the same time; or } i\equiv j \end{cases}$$ build adjacency matrix

$$A_{i,j} = A_{j,i} = \begin{cases} 1 & W_{i,j}=1 \\ 0 & W_{i,j}\neq 1 \end{cases}$$

In Fig. 2.， P_1 and P_4 are in the gap, and belong to the same triangle mesh $\Delta P_1P_4P_6$, so $W_{1.4}=1$, $A_{1,4}=1$, P_1 and P_4 are interconnected in the gap. P_4 and P_6 are also in the gap, but P_4 and P_6 belong to triangular mesh $\Delta P_1P_4P_6$ and $\Delta P_4P_5P_6$, so $W_{1.4}=2$, $A_{1,4}=0$, P_4 and P_6 are disconnected.

For each node P_i in adjacency matrix，degree of the node $\mathrm{TD_i} = \sum_{j=1}^{n} A_{i,j}$, edge $e = \frac{1}{2}\left\{\sum_{i=1}^{n} TD_i\right\}$, generally, after triangle FEM, amount of gap group is less than the amount of triangle mesh, $e << n^2$, adjacency matrix is a

sparse matrix. In order to improve the storage efficiency, adjacency list data structure in physical is applied. A good traversing algorithm is for whole data set. There are two graph traversing methods, DFS（Depth First Search）and BFS(Breadth First Search). There are n nodes, e edges in the graph, and store the sparse matrix by adjacency list. In DFS algorithm, all adjacency nodes of one node P can be found along the adjacency list, time cost to search the edge is O(e), as there are 2e nodes, all nodes are visited only once, time complexity of graph traversing is O(n+e). In BFS algorithm, time complexity of graph traversing is $TD_1 + TD_2 + ...TD_n = O(e)$, TD_i is the degree of node P_i. BFS algorithm is better in adjacency matrix traversing.

4.2. First Classification

For undirected connected graph, composed of gaps in whole ship triangle mesh, a marked array T is used to record the visited gap node. When P_i, P_j are interconnected ($A_{i,j} = 1$), P_i, P_j belong to a same gap group G. In BFS algorithm, for marked array T and gap group G: If a node has already been visited, this node is part of a connected component, which is the existing gap group $G_1, G_2, ...G_s$. If not, execute graph traversing from this node, another connected component, a new gap group G_t can be found. From the node P_1, visit the adjacency nodes $P_2, P_3, ...P_n$ in turn, $P_2, P_3, ...P_n$ are nodes that have not been visited. Mark the visited node in array T, if the node belong to the existing gap group G_s, add the node, if not, and add a new gap group G_t. Repeat this procedure until all the nodes have been visited. In Fig.3, the whole gap is divided into four gap groups, as show in Fig.4 and Fig.5.

According to TD_i, classify the gap group into two type: $G_s \in G_{ok}$（P_i is a node in G_s, and degree of each node is less than 4, $\mathrm{TD}_j < 4$）; $G_t \in G_{regroup}$（P_j is a node in G_t, and there is a node which degree is more than 4, $\mathrm{TD}_j \geq 4$）.Type I has one polygon gap (Fig. 4.), type II has two or more polygon gaps (Fig.5).

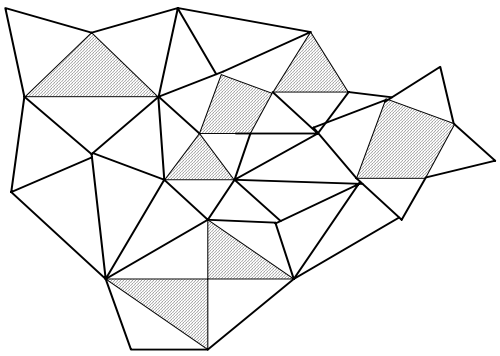


Fig. 3. Triangle elements with gap

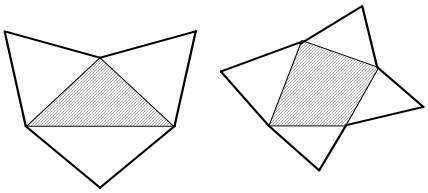


Fig. 4. Gap team had been grouped completely

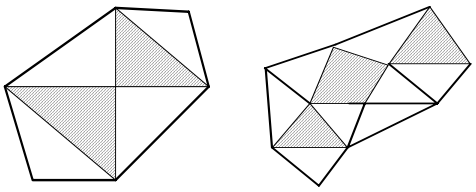


Fig. 5. Gap team need to be regrouped

4.3 Deeper Classification

One gap group has one polygon for triangle mesh mending use, so $G_{regroup}$ is need to be regrouped. In Fig.2, it is necessary to divide group $G = \{P_1, P_2, P_4, P_6, P_8\}$ to two new groups $G_1 = \{P_1, P_2, P_4\}$ and $G_2 = \{P_1, P_6, P_8\}$, the realization is as the following:

Step1. Build new adjacency matrix $A_{regroup[i,j]}$ for all gap nodes in $G_{regroup}$.

Step2. Find all the articulation nodes of $TD_i \geq 4$ in $A_{regroup[i,j]}$.

Step3. Delete all articulation nodes and its adjacency edges, build adjacency matrix $A_{revers[i,j]}$, represented of two or more connected components,

Step4. For the connected graph represented of $A_{revers[i,j]}$, using first classification method, divide the connected components individually.

Step5. According the connectivity of $A_{regroup[i,j]}$, add articulation nodes to the gap group of step4.

After regrouping, result of Fig. 5. is as the following:

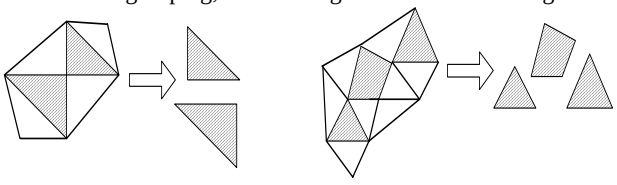


Fig. 6. Regroup gap process

5. GAP MENDING

5.1 Gap Mending of Three Nodes

Max-min angle criterion is used to mend mesh gap of three nodes, convex quadrilateral region, made of two adjacent triangles, connect with the diagonal line to get an optimal triangle shape. Get max edge $e_{\max}$, min edge $e_{\min}$, inner corner $Angle_{\max}$, $Angle_{\min}$, and related element $Elm_{e_{\max}}$, $Elm_{e_{\min}}$ from triangles set $Elm\{Elm_1, Elm_1, ...Elm_m\}$

1. If $e_{\min} < e_{\max}/5$, *and* $e_{\min}$ <*mesh size 5*, delete min edge $e_{\min}$ and it's related element $Elm_{e_{\min}}$, make two nodes of the $e_{\min}$ equivalence (Fig. 7.).

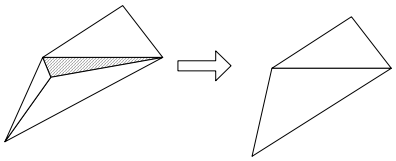


Fig. 7. Delete small element and make nodes equivalence

2.If inner angle $Angle_{\max} > 135^0$ *or* $Angle_{\min} < 15^0$, delete max $e_{\max}$ and it's related element $Elm_{e_{\max}}$, four nodes gap will be created. By connecting the third node of $Elm_{e_{\max}}$ and the third of the gap, two new angles will be created.

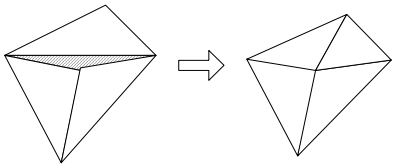


Fig. 8. Delete large element and create triangle mesh

5.2. Gap Mending of more than three Nodes

First, checking the embedded triangle element by traversing all the gap nodes, embedded element are composed of gap nodes. Delete the embedded triangle element, so that the gap is composed of border edges.

At present, curvilinear surface mesh rebuilding method are classified into two types, based on surface and volume. A method, composing a separate curved surface of Implicit Approximation by rebuilding topology and geometry at the same time is provides by Hoppe. Nuid, draw on Hoppe's ideas, provide a space triangulation method based on 3D scattered data [7]. In this paper, Nuid's algorithm is improved for triangular mesh rebuilding. The initial triangle may be an arbitrary triangle $Elm_{e_{start}}$, which is related to the gap. Add others triangles start with the edge $e = (P_1, P_2)$ of $Elm_{e_{start}}$, define k neighborhood Nbhd (e) = Nbhd (P_1) Y Nbhd (P_2), the best node is in Nbhd (e). Algorithm of finding the best node is as the following:

Step1. Nodes of Nbhd (e) are projected micro tangent plane $(P-c)\cdot n_c = 0$, n_c is driving poles normal of current edge's neighborhood, set $\cos_{\min} = 2, n_{ref} = n_c$

Step2. Get untested node P_3 from quasiperiodical point set of Nbhd (e), and calculate cosine of the angle, $\cos\angle P_1P_3P_2 = \dfrac{(P_1^{'} - P_3^{'})(P_2^{'} - P_3^{'})}{|(P_1^{'} - P_3^{'})(P_2^{'} - P_3^{'})|}$, if $\cos\angle P_1P_3P_2 < \cos_{\min}$ then $\cos_{\min} = \cos\angle P_1P_3P_2$, P_3 is the best node, create new triangle element $P_1P_2P_3$. If $\cos\angle P_1P_3P_2 < \cos_{\min}$ is not right for all neighborhood, change another edge e and restart.

6. CONCLUSION

Application of FEM analysis in ship structure is developing. The logical relationship of the model data is frequently lost in the transformation between the design and analysis software. The holes, slots and small curved surface caused by the failure of transformation will create triangle mesh gap in FEM analysis model. In order to mend the gaps, associating with the existing triangle elements mending methods, we provide: (1) Group the gaps by adjacency matrix. First, classifying the gaps using undirected connected graph data structure and BFS algorithm, then, using adjacency matrix and adjacency matrix skim off articulation nodes to execute deeper classification. BFS algorithm and marked array is applied to adjacency matrix traversing (2) Mend and rebuild the triangle element using comparable methods for the gaps. After these processing methods, FEM analysis for strength calculation can be completed in some details of the whole ship model, such as the connecting component of the cabin.

7. REFERENCE

[1] Q.Q.Cheng, S.C.Zhu, "Research on numerical technique of whole large container ship by FEM", *Journal*, Journal of Ship Mechanics, 2006,10(1): 80-91.

[2] E.R.Qi, W.C.Cui, "Nonlinear finite element analysis of ultimate strength of damaged ship hulls ", *Journal*, Journal of Ship Mechanics, 2005,9(5): 83-91.

[3] Anglada Marc Vigo, "An improved incremental algorithm for constructing restricted Delaunay triangulations", *Journal*, Computer & Graphics, 1997,21(2): 215-233.

[4] W.L.Chen, S.Zhang, et al, "Research on the Algorithm of Hole Repairing for Finite Element Mesh", *Journal*, Chinese Journal of Computers 2005,28(6): 1068-1071.

[5] S.J.Xiao, S.S.Zhang, et al, "Integrated Optimization of Triangle Net for Scattered Data Points and Its analysis", *Journal*, China Mechanical Engineering 2002, 13(19): 1666- 1668.

[6] S.Xu, J.Y.Wang, "Adaptive Triangulation of Curved Surface", *Journal*, Journal of Computer Aided Design & Computer Graphics, 2000,12(4): 267-271.

[7] L.B.Du, J.C.Sun, et al, "Research on algorithm of surface with hole in reverse engineering", *Journal*,

Shandong Science 2005,18(5): 58-66.

Steganography Based On Self-Organizing Competitive NNS And HVS *

Z.W.Kang1, J.Liu1, Y.G.He2

1.School of Computer and Communication, Hunan University, Changsha 410082,China;

2. School of Electrical and Information Engineering, Hunan University, Changsha 410082, China

Email: hn_zwkang@126.com

ABSTRACT

In order to provide larger capacity of hidden secret data and imperceptibility of stego-image, a novel steganographic method based on self-organizing competitive neural networks (NNs) and human vision system (HVS) was presented. First, self-organizing competitive NNs based on competitive learning was trained according to contrast and texture sensitivity, the NNs trained could thus distinguish the pixels in less sensitive areas from those in more sensitive areas. Then the NNs trained was exploited to estimate the degree of sensitivity of pixels so that more secret data could be embedded in less sensitive areas of vision. From the experimental results, compared with the Four-sided Side Match method, the proposed method hides much more information and maintains a good visual quality of stego-image as well.

Keywords: Steganography; HVS; Side match; Sensitivity; Self-organizing competitive NNs.

1. INTRODUCTION

Steganography is a newly-developed technique of information hiding. The most commonly used steganographic method is LSB (least significant bit) [1,2], which replaces the least significant bits of cover image with secret information to be embedded. Generally speaking, a good steganographic technique should have both good visual imperceptibility and a sufficient capacity of hidden secret data [3]. Although LSB maintains a good visual quality of stego-image, it hides a little information. Considering the drawback of LSB, some methods begin to take account of the visual identity that human eyes are insensitive to edged and textured areas when embedding secret information, such as BPCS(bit-plane complexity segmentation) [4], PVD(pixel-value differencing) [5], MBNS(multiple-base notational system) [3] and Side Match [6]. With these methods, more secret information is embedded in edged areas, less in smooth areas. The capacity of embedded information is thereby greatly improved while the quality of visual imperceptibility is maintained.

However, it is hard to decide whether a pixel is in less sensitive areas or not. In this paper, we propose a novel steganographic method accordingly, in which we train self-organizing competitive neural networks [7] (NNs) based on contrast and texture sensitivity to distinguish pixels in less sensitive areas from those in sensitive areas. As a result, the NNs trained is the secret key. Then, we use the NNs trained to classify pixels and select those in less sensitive areas to embed more secret data. On the receiving side, with this method, the secret information can be extracted from stego-image without referencing the original image.

This paper is organized as follows. In the next section, we give a brief introduction to self-organizing competitive neural networks. Then in section 3, we illustrate the algorithm proposed in detail, thereafter the experimental result is given in section 4. The last part of this paper is the conclusion.

2. THE SELF-ORGANIZING COMPETITIVE NNS

Self-organizing competitive NNs [7] is made up of input layer and competitive layer. Fig.1 shows the structure of self-organizing competitive NNs. And *N* and *M* stand for input layer and competitive layer of the neuron, respectively. Where, w_{ji} are the weights, *i*=1,2…*N*, *j*=1,2…*M*.

$$\sum_i w_{ji} = 1 \tag{1}$$

H learning sample is $p_k = (p_1^k, p_2^k ..., p_n^k)$, and the output sample corresponding to p_k is $a_k = (a_1^k, a_2^k ..., a_n^k)$, *k* =1,2…*H*. The learning rule is as follows:

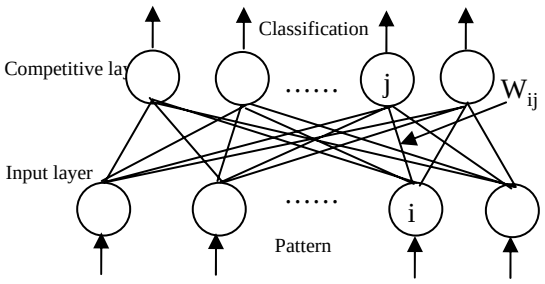


Fig. 1. The structure of self-organizing competitive NNs

Step 1: Give a random value between 0 and 1 to w_{ji}, *i*=1,2…*N*, *j*=1,2…*M*. where w_{ji} satisfy Eq.(1).

Step 2: Choose a sample p_k from *H* learning sample as input of NNs.

Step 3: Input value S_j of competitive layer is computed as

$$S_j = \sum_i w_{ji} x_i \tag{2}$$

Step 4: According to the principle that the winner is king, the neuron whose S_j is maximal is winning neuron, and its output value is 1. Otherwise, its output value

* This work was supported by the National Natural Science Foundation of China (No. 50277010).

is 0.

Step 5: The weights connecting to winning neuron is adjusted as $w_{ji} = w_{ji} + \Delta w_{ji}$ (3)

where $\Delta w_{ji} = \eta(\frac{x_i}{m} - w_{ji})$ (4)

where η is learning rate, m is the number of element 1 in p_k, while others remain unchanged.

Step 6: Not choose another learning sample, return to step 3 until *H* learning sample is provided to NNs.

Step 7: Return to step 2, until the change of weights is small.

3. PROPOSED METHOD

Based on visual identity of human eyes, the proposed method train the self-organizing competitive NNs with the contrast and texture sensitivity as input, the NNs trained hence tell the degree of sensitivity of pixels. Thus the amount of information to be embedded is decided accordingly. Following is the detail.

3.1 Training Self-organizing competitive NNs

Some pixels with neighboring pixels (upper, left, right, bottom) are randomly selected from several images. Given an input pixel P_X with gray value g_x, let g_u, g_l, g_r and g_b be the gray values of its upper P_U, left P_L, right P_R and bottom P_B pixels, respectively. The texture sensitivity, the variance of 4 neighboring pixels，is computed as

$$T = \sqrt{\sum_{i=u,l,r,b} (g_i - m)^2} \quad (5)$$

where， $m = \frac{1}{4}(g_u + g_l + g_r + g_b)$

The contrast sensitivity, the maximal gray value distance of 4 neighboring pixels，is computed as

$$C = \max_{i=u,l,r,b}(g_i) - \min_{i=u,l,r,b}(g_i) \quad (6)$$

Above 2 elements constitute input vector $V\{T, C\}$ of pixel P_X, where，T and C denote texture sensitivity and contrast sensitivity, respectively. Vector V as input, self-organizing competitive NNs is trained.

3.2 Embedding Method

As shown in Fig. 2., the white pixels are used for embedding secret data, while the black pixels remain unchanged. Following is the secret data hiding process in detail:

Step 1: The texture sensitivity T and the contrast sensitivity C of P_X are computed as Eq. (5) and Eq. (6).

Step 2: The vector $V\{T, C\}$ is inputted, and whether this pixel is in less sensitive areas or not is obtained.

Step 3: If this pixel is in less sensitive areas, 4 bits are embedded, namely, $L = 4$. Otherwise, 2 bits are embedded, that is $L = 2$

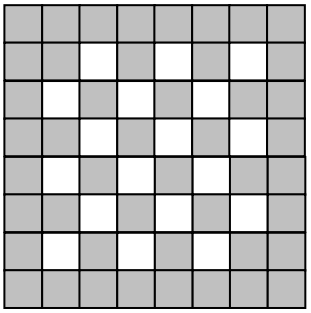


Fig. 2. The sampling arrangement for proposed method

Step 4: Transform L bits binary secret data into decimal value d.

Step 5: Modify the pixel value g_x to embed secret data d.

$$g_x^* = \arg\min_{v \in [0,255], \mathrm{mod}[v, 2^L] = d} |v - g_x| \quad (7)$$

3.3 Extracting Method

As the process of embedding, extract secret data from white pixel and compute the texture sensitivity T and the contrast sensitivity *C* as Eq. (5) and Eq. (6). If this pixel is in less sensitive areas, 4 bits are extracted, $L = 4$. Otherwise, 2 bits are extracted, $L = 2$. Then, the value of embedding data d is computed as Eq. (8). Finally, transform decimal value *d* into *L* bits binary secret data.

$$d = \mathrm{mod}(g_x^*, 2^L) \quad (8)$$

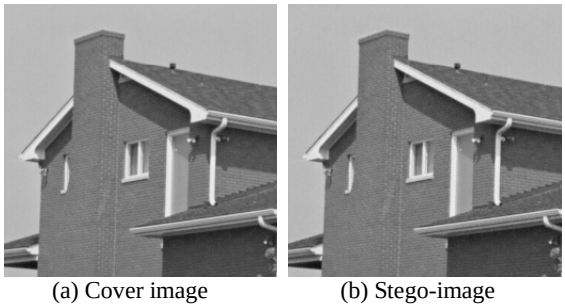
4. Experimental results

In order to validate the characteristics of larger capacity and imperceptibility of stego-image, this paper chooses 8000 pixels randomly from several images as training data, and trains NNs with 0.01 as learning rate and 5 as epoch. Then we choose fake stochastic sequence as secret information, and PSNR as the measurement of imperceptibility.

Fig. 3. and Fig. 4. present the experimental results using standard image of Lena and House. Fig.3(c) and Fig.4(c) show the difference images between the stego-images and the cover images (with the differences of gray values being scaled 32 times), we can see that major changes lie in texture areas and edges of the image, therefore, good visual imperceptibility is obtained. Finally, eight standard test images (256*256) are chosen for experiment. And the comparison of the message capacity and the PSNR value of two methods: four-side Side Match and the proposed one is shown in table 1.It is seen that the proposed method has about a hiding capacity 1.52~2.06 times as much as that of four-sided side match. Although the PSNR value drops 4.6db at most, the PSNR value is higher than 38db with which human eye could not notice the decline of quality of stego-image.

(a) Cover image (b) Stego-image (c) The difference image

Fig. 3. Cover image, stego-image and the difference image of Lena



(a) Cover image (b) Stego-image (c) The difference image

Fig. 4. Cover image, stego-image and the difference image of House

Table 1. Comparison of results of the proposed and Four-sided Side Match methods

Cover image 256*256	Four-sided Side Match		Proposed method	
	Capacity (bit)	PSNR (db)	Capacity (bit)	PSNR (db)
Lena	45571	42.4508	91700	41.2581
Bridge	70172	39.5899	117022	38.6793
House	40115	47.2834	82826	42.6529
Baboon	75677	37.8653	115040	38.8847
Man	52141	42.4090	99044	40.3554
Plane	45943	42.3195	87616	41.8214
Peppers	48936	42.6682	90550	41.3981
Couple	64904	39.6379	105856	39.6660

5. CONCLUSIONS

This paper presents a steganographic method based on self-organizing competitive NNs and HVS. In the proposed steganography, whether the pixel is in less or more sensitive areas is decided by NNs trained. As more data are embedded in less sensitive areas which can tolerate more changes, the method provides a good imperceptibility with a large quantity of embedded data. Simulation experiments show that the proposed method has much greater hiding capacity than Four-sided Side Match [6], maintaining a good visual quality of stego-image. On the other hand, the amount of information carried by individual pixels is decided by NNs trained, which is a secret key. Therefore, any third party will be unable to extract the embedded data.

6. REFERENCES

[1] Bender W, Gruhl D et al, "Techniques for Data Hiding", IBM System Journal, 1996, 35(3):313~336.

[2] Petitcolas F A P, Anderson R J et al, "Information Hiding—A Survey", Proc. IEEE, 1999, 87(7): 1062-1078.

[3] Zhang X, Wang S, "Steganography Using Multiple-Base Notational System and Human Vision Sensitivity", IEEE Signal Processing Letters, 2005, 12(1):67-70.

[4] Kawaguchi E, Eason R O, "Principle and Application of BPCS-Steganography", In Proceedings of SPIE: Multimedia Systems and Applications, 1998, 3528: 464-472.

[5] Wu D C, Tsai W H, "A Steganographic Method for Images by Pixel-Value Differencing", Pattern Recognition Letters, 2003, 24(10):1613-1626.

[6] Chin-Chen Chang, Hsien-Wen Tseng,"A steganographic method for digital images using side match", Pattern Recognition Letters,2004, 25(12):1431–1437

[7] Rosenblatt, F., "The Perceptron: A probabilistic model for information storage and organization in the brain", Psychological Review, 1958, 65(1):386-408.

The Bilinear Nondestructive Algorithm of Image Edge Detection *

Pei You Han
College of Computer Science and Information Engineering, Zhejiang Gongshang University
Hangzhou, Zhejiang 310012, China
Email: happyyou@mail.zjgsu.edu.cn

ABSTRACT

According to generalized fuzzy sets theory, the linear generalized fuzzy operator of contrast fuzzy enhancement was presented, and then the bilinear nondestructive algorithm of edge detection was given. First translated the normal space of grey image to corresponding generalized fuzzy space using left semi-trapezoid fuzzy distribution function, second proceed fuzzy enhancement of contrast to generalized fuzzy sets, third translated generalized fuzzy sets to normal sets, at last the edge was extracted in normal sets. Using this algorithm to extract edge is of fast velocity and of good effect, and exceeded the method listed in literature on multi-index.

Keywords: Edge Detection, Generalized Fuzzy Sets, Linear Generalized Fuzzy Operator, Fuzzy Enhancement.

1. INTRODUCTION

Theory of fuzzy sets, which was a soft compute tool characterizing the un-precise or uncertain question and researching the incomplete and inaccurate question of information system, was established by L. A. Zadeh [5] in 1965. Chen Wu fan, in 1985, developed generalized fuzzy sets (*GFS*) and further advanced a new algorithm of image edge detection, namely generalized fuzzy operator (*GFO*).

In the field of image processing, edge detection, normally applied in two great subjects [1,2…23,24]: image segmentation and image analysis, is an important branch. Edge detection has many methods to realize, mostly based on theory of grey un-continuity, but more and more scholars emphasize technique based on theory of fuzzy sets. Generally a good image system should be compatible with vision mechanism, so we have expected a model, which can describe the characteristic vision. Hence theory of fuzzy sets, an effective compute tool in artificial intelligence, pattern recognition, image processing, and image analysis etc., was introduced into algorithm of image edge detection increasingly. Pal [1,2] and other person offered a fuzzy algorithm of image edge detection which has obtained good result in *X*–ray image edge detection, but led to the large amount of compute and lost edge information of low grey value of image because of changing normal space sets into generalized fuzzy space sets by using fuzzy membership of power function, meanwhile using nonlinear transform in enhancing processing.

Based on theory of generalized fuzzy sets, this article gave the linear generalized fuzzy operator (*LGFO*) of fuzzy enhancement of contrast among successive region, and then put forward the bilinear nondestructive algorithm of image edge detection. First translated the normal space of grey image to corresponding generalized fuzzy space using left semi-trapezoid fuzzy distribution, second proceed fuzzy enhancement of contrast among successive region to GFS, and then translated GFS to normal space, at last the edge was extracted in normal sets. A great deal of instance indicated that using the algorithm of this paper to extract the edge is of fast velocity and of good effect, exceeded the method listed in literature [1,2,3,4] on multi-index. Using dynamic transform algorithm method of *K–L* offered by Y. Ohta [9] and others, the algorithm of this paper was also adapted to color image.

*Foundation item: Supported by the Natural Science Foundation of Shaanxi Province (HJ04033). Biographies: HAN Pei you (1964 –), male, doctor, associate professor, engages in database system, pattern recognition and image processing.

2. GENERALIZED FUZZY SETS AND LINEAR GENERALIZED FUZZY OPERATOR

As expansion of the theory of classical sets, the new soft compute method - generalized fuzzy sets, offered a new soft research method for artificial intelligent, compute vision, pattern recognition, and image processing.

We give concept of generalized fuzzy sets and linear generalized fuzzy operator as follow:

Definition 1: A fuzzy sets *A* of Universe *U* is defined by a membership function:

$$\mu_A(u) : U \rightarrow [0, 1]$$

where $\mu_A(u)$ denotes membership degree of element u belong to set *U*.

A fuzzy sets *A* is denoted by $\mu_A(u)$.

Definition 2: A generalized fuzzy sets *A* of Universe *U* is defined by a generalized membership function:

$$\mu_{GA}(u) : U \rightarrow [-1, 1]$$

where $\mu_{GA}(u) \in [-1, 0]$ denotes membership degree of element u not belonging to set *U* completely, $\mu_{GA}(u) \in (0, 1)$denotes membership degree of element u completely belonging to set *U*, $\mu_{GA}(u) = 0$ denotes boundary point of set *A* of *U*.

A generalized fuzzy sets *A* is denoted by $\mu_{GA}(u)$.

Definition 3: Linear generalized fuzzy operator is the transform that translate a generalized fuzzy sets *A* into a normal sets *A'*, denoted by *A' = LGFO* (*A*).

The definition of the function *LGFO* is Eq. (1).

According to definition 3, it is not difficult to prove that *LGFO* has properties as follow:

Property 1: For every $-1 \leqslant \mu_A(x) \leqslant 1$, then formula(1) is continuous and linear.

Property 2: For every $-1 \leqslant \mu_A(x) \leqslant 1$, then $0 \leqslant LGFO(\mu_A(x)) \leqslant 1$, *viz*. Translate generalized fuzzy sets into normal fuzzy sets.

Property 3: For every $-1 \leqslant \mu_A(x) \leqslant 0$ or $r \leqslant \mu_A(x) \leqslant 1$, then $\mu_A(x) \leqslant \mu_{A'}(x)$. *viz*. pixel value of image was increased in region $-1 \leqslant \mu_A(x) \leqslant 0$ or $r \leqslant \mu_A(x) \leqslant 1$.

Property 4: For every $0 \leqslant \mu_A(x) \leqslant r$, then $\mu_{A'}(x) \leqslant \mu_A(x)$. *viz*. pixel value of image was increased in region $0 \leqslant$

$\mu_A(x) \leqslant$ r.

$$\mu_{A'}(x) = LGFO(\mu_A(x)) =$$

$$\begin{cases} -\frac{1}{n}\mu(x) - \frac{1}{n} + 1 & -1 \le \mu(x) < -r \\ -\frac{1-r}{r}\mu(x) - \frac{r-1}{n} + r & -r \le \mu(x) < -\frac{1}{n}r \\ -n\mu(x) & -\frac{1}{n}r \le \mu(x) < 0 \\ \frac{1}{n}\mu(x) & 0 \le \mu(x) < \frac{1}{2}r \\ \mu(x) - \frac{n-1}{2n}r & \frac{1}{2}r \le \mu(x) < \frac{2n-1}{2n}r \\ n\mu(x) - (n-1)r & \frac{2n-1}{2n}r \le \mu(x) < \frac{2n-1}{2n}r + \frac{1}{2n} \\ \mu(x) + \frac{n-1}{2n}(1-r) & \frac{2n-1}{2n}r + \frac{1}{2n} \le \mu(x) < \frac{1}{2}(r+1) \\ \frac{1}{n}\mu(x) - \frac{1}{n} + 1 & \frac{1}{2}(r+1) \le \mu(x) \le 1 \end{cases} \quad (1)$$

where $r \in (0,1)$, n = 1, 2, 3, … is an adjustable parameter.

From property 1 to property 4 we can see that formula (1) can achieve the effect of enhancing image contrast. This operator is linear, as a result, the algorithm is more effective in detection and less complexity than the methods listed in literature [1,2,3,4].

3. LINEAR GENERALIZED FUZZY TRANSFORM

In order to proceed enhancement process to grey image $I = (x_{ij})$ ($x_{ij} \in \{0,1,\ldots,255\}$, $i \in \{1,2,\ldots,m\}$, $j \in \{1,2,\ldots,n\}$) , we need to transform the image to generalized fuzzy sets $P=(p_{ij}) \in [-1,1]$. In this paper, linear left semi-trapezoid fuzzy distribution is used as the generalized membership transform of image.

The linear generalized membership transform of image is:

$$p_{ij} = LT(x_{ij}) = \frac{x_{ij} - M}{x_{max} - M} \quad (2)$$

where $0 < M \leqslant (x_{max} - x_{min})/2$ is adjustable parameter, x_{max} and x_{min} are respectively the maximum and minimum of image pixel value. It is not difficult to prove that the generalized fuzzy set $P = (p_{ij})$ transformed from linear generalized membership transform $LT(x_{ij})$ meets:

$$P = (p_{ij}) \in [-1, 1] .$$

4. BILINEAR FAST DETECTIVE ALGORITHM

In conclusion, as to image I, bilinear fast edge detective algorithm is described as follow:

Step 1: Grey processing of color image.

If image I is color image, namely $I = (R_{ij}, G_{ij}, B_{ij})$ (R_{ij}, G_{ij}, $B_{ij} \in \{0, 1, \ldots, 255\}$, $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$), then need to transform bitmap RGB to grey image $I = (x_{ij})$.

The concrete method is:

$$x_{ij} = 0.29900 * R_{ij} + 0.58700 * G_{ij} + 0.11400 * B_i \quad (3)$$

where $x_{ij} \in \{0,1,\ldots,255\}$, ($i \in \{1,2,\ldots,m\}$, $j \in \{1,2,\ldots,n\}$) represent the grey value of every pixel; R_{ij}, G_{ij}, B_{ij} represent respectively the component color values of red, green, blue of each pixel.

Step 2: Generalized processing of grey image.

For grey image $I = (x_{ij})$ ($x_{ij} \in \{0, 1, \ldots, 255\}$, $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$) , using linear generalized membership transform $LT(x_{ij})$ indicated in formula(2) , transform grey image $I = (x_{ij})$ to generalized fuzzy sets $P = (p_{ij}) \in [-1,1]$.

Step 3: Enhancement processing of generalized fuzzy set.

For generalized fuzzy sets $P = (p_{ij})$, using linear generalized fuzzy operator (*LGFO*) indicated in formula (1) , transform generalized fuzzy sets $P = (p_{ij}) \in [-1,1]$ to normal fuzzy set $P' = (p'_{ij}) \in [0,1]$. In fact, the procedure has function of fuzzy enhancement for image.

Step 4: Grey processing of normal fuzzy set.

For normal fuzzy set $P' = (p'_{ij})$, use the following reverse transform of linear generalized membership to transform normal fuzzy sets $P' = (p'_{ij})$ to enhance grey image, $I' = (x'_{ij})$ ($x'_{ij} \in \{0,1,\ldots,255\}$, $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$) .Where linear generalized membership reverse transform $RLT(p'_{ij})$ is:

$$x'_{ij} = RLT(p'_{ij}) = LT^{-1}(p'_{ij}) = p'_{ij} * (x_{max} - M) + M \quad (4)$$

Step 5: Edge extraction.

At last, use "MIN" or "MAX" [10] to extract the edge of image.

Bilinear fast edge detective algorithm for image can be expressed as Fig. 1:

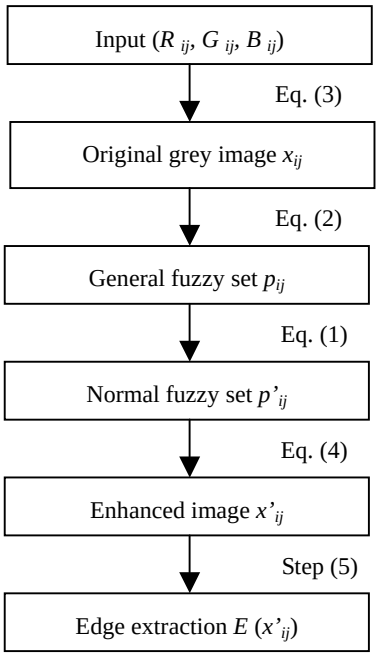


Fig. 1. Bilinear fast edge detective algorithm

5. APPLICATION EXAMPLES OF BILINEAR DETECTIVE ALGORITHM AND EXPERIMENT COMPARISON

Use the algorithm presented in this paper to enhance contrast and extract edge for original image Fig. 2, and the selection of each parameter is as Fig. 3 ~ Fig. 7. Where:

Fig. 2 indicates standard test image,

Fig. 3 indicates the edge extracted by Pal operator,

Fig. 4 indicates enhanced processing image with r = 0.6, M=98, n=5 of *LGFO* in this paper,

Fig. 5 indicates the edge extracted with r=0.6, M=98, n=5 of *LGFO*,

Fig. 6 indicates enhanced processing image with r=0.5, M=60, n=10 of *LGFO*,
Fig. 7 indicates the edge extracted with r=0.5, M=60, n=10 of *LGFO*.

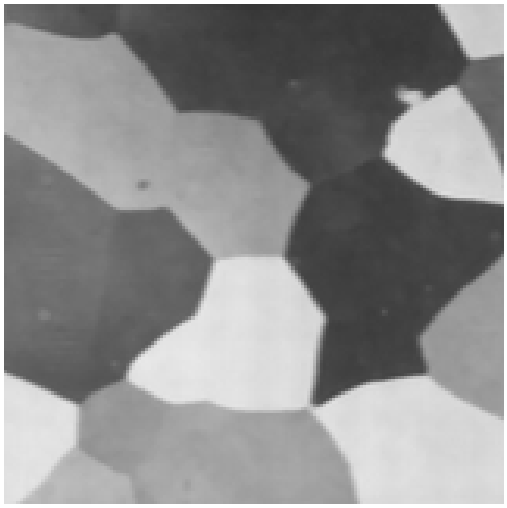


Fig. 2. Original image

Fig. 3. *Pal* operator

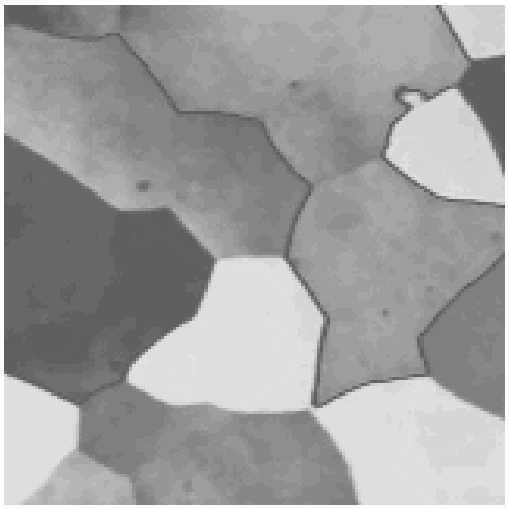


Fig. 4. *LGFO*(r=0.6, M=98, n=5)

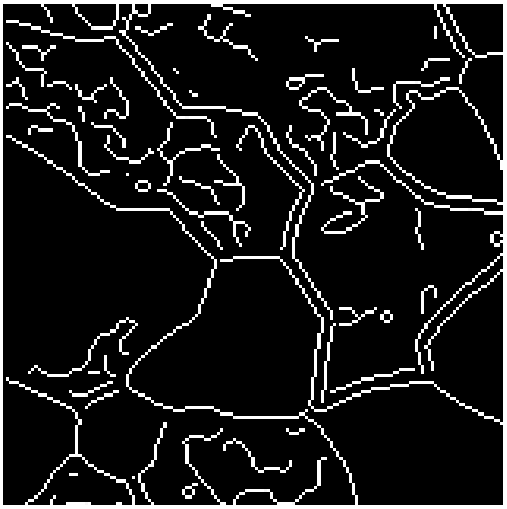


Fig. 5. *LGFO*(r=0.6, M=98, n=5)

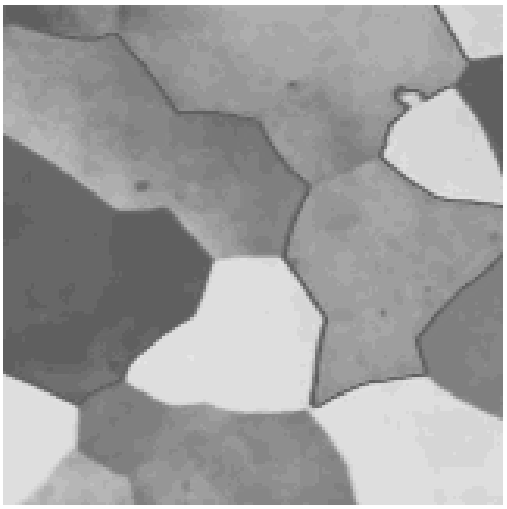


Fig. 6. *LGFO*(r=0.5, M=60, n=10)

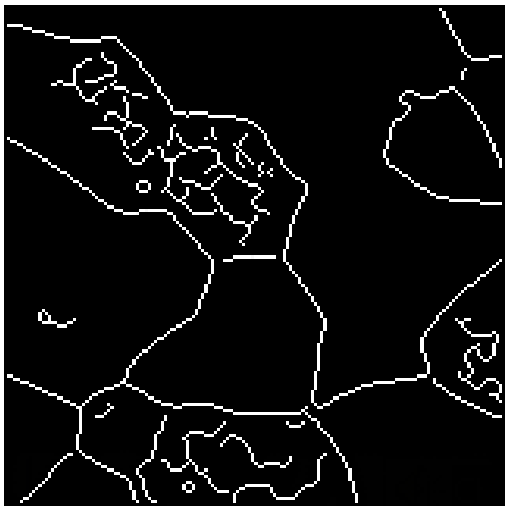


Fig. 7. *LGFO*(*r*=0.5, *M*=60, *n*=10)

Experiment comparison of edge detective algorithm as follow (Table 1):

Table 1. Experiment Comparison

Algorithm	Pal	Chen Wufan	Wang hui	This paper
Complexity	$O(x^{5/2})$	$O(x^{5/2})$	$O(x^{5/2})$	$O(x)$
Speed(s)*	6.6538	4.5826	2.2646	0.8375
Dynamic detective	no	no	no	yes (r, M, t)
Linear	no	no	no	yes
Continuity	no	no	no	yes
Lossless	loss	loss	loss	lossless

6. CONCLUSIONS

This article, based on theory of *GFS*, gave new generalized fuzzy operator of fuzzy enhancement of contrast image successive region, then put forward the bilinear nondestructive algorithm of image edge detection. Using the algorithm of this article to detect image edge has following virtues:

(1) Generalized fuzzy processing of image, using left semi-trapezoid fuzzy distribution linear transform *LT* (x_{ij}) (step 2).

(2) Generalized fuzzy enhancement processing of image, using linear generalized fuzzy operator (*LGFO*) (step 3).

(3) Normal fuzzy sets grey processing, using linear transform *RLT* (p'_{ij}) (step4).

(4) The bilinear fast algorithm of image edge detection is double linear lossless linear transform, so in image processing the information of high frequency and low frequency are lossless and can be restored without loss.

(5) Through adjusting the value of adjustable parameters *r*, *n* and *M*, we can extract the more detailed edge information or may ignore the useless edge information, so as to meet the requirements a variety of image processing.

(6) Using the method of dynamic *K-L* transform [9], the algorithm of this paper is also adapted to the color image.

A great deal of instance indicated that using the algorithm of this paper to extract the edge is of fast velocity and of good effect, exceeded the method listed in literature [1,2,3,4] on multi-index.

In addition, the algorithm of this paper adapts to many kinds of image. For the sake of the randomicity and complexity of image texture structure, how to select the value of adjustable parameters r, n and M to extract ideal image edge, which can meet the requirements of different usages, is still a problem not settled. So to exploit an adaptive algorithm for the bilinear lossless of edge detection is the main research goal for image processing in the future.

7. REFERENCES

[1] S.K.Pal, R.A.King, On edge detection of X-ray images using fuzzy sets, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983, 5(1):69-77.

[2] S.K.Pal, R.A.King, Image enhancement using smoothing with fuzzy sets, IEEE Transactions on System, Man, and Cybernetics, 1981,11（7）：494 - 501.

[3] W.F. Chen, X.Q. Lu, “New algorithm of edge detection of color image”, Science in china (A), 1995, 25（2）：219 -225.

[4] H. Wang, J.H. Zhang, “An algorithm of edge detection based on fuzzy enhancement of contrast among successive regions”, Acta electronica sinica, 2000, 28（1）: 45-47.

[5] L. A. Zadeh, “Fuzzy sets”, Information and control, 1965, 8（3）: 338 - 353.

[6] Z. Pawlak, “Rough sets and fuzzy sets”, Fuzzy sets and systems, 1985, 17 - 99.

[7] S. Nanda, “Fuzzy rough sets”, Fuzzy sets and systems, 1992, 45- 157.

[8] K. Chakrabarty, R. Biswas, et al, Fuzziness in rough sets [J], Fuzzy sets and systems, 2000, 110~247.

[9] Y. Ohta, “Color information for region segmentation”, Computer graphics and image processing, 1980, 13（3）：222-241.

[10] Y.Nakagawa, A.Rosebfeld, “A note on the local min and max operations in digital picture processing”, IEEE Transactions on System, Man, and Cybernetics, 1978, 8(8): 632 - 635.

[11] R.L.Queiroz, Z. Fan, Optimizing block - thresholding segmentation for multilayer compression of compound images [J], IEEE Transactions on Image Processing, 2000,9(9): 1461-1471.

[12] H.D. Cheng, X.H. Jiang, et al, “Color image segmentation: advances and prospects”, Pattern Recognition, 2001, 34（12）: 2259 -2281.

[13] H.D. Cheng, X.H. Jiang, et al, “Color image segmentation based on homogram thresholding and region merging”, Pattern Recognition, 2002,35(2): 373-393.

[14] S. T. Acton, D. P. Mukherjee, “Scale space classification using area morphology”, IEEE Transactions on Image Processing, 2000, 9（4）: 623-635.

[15] P. K. Saha, J. K. Udupa, “Optimum image thresholding via class uncertainty and region homogeneity”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23（7）: 689 - 706.

[16] P. L. Rosin, “Unimodal thresholding”, Pattern Recognition, 2001, 34（11）: 2083 -2096.

[17] G. Gagaudakis , P. L. Rosin, “Incorporating shape into histograms for CBIR”, Pattern Recognition, 2002, 35（1）: 81- 91.

[18] N. D. Venkata, B. L. Evans, “Adaptive threshold modulation for error diffusion half-toning”, IEEE Transactions on Image Processing, 2001, 10（1）: 104 - 116.
[19] P. Z. Zeng, S.Y. Yu. Fuzzy mathematics and its application. Wu han, Wu han university press, 2002.
[20] Y. Cui. Image processing and analysis. Beijing, Science press, 2002.
[21] Y.J. Zhang. Image segmentation. Beijing: Science press, 2001.
[22] X.S. Zhao, S.Z. Chen. Segmentation of Letter Images [J]. Journal of computer-aided design and computer graphics, 2002, 14(7): 701 - 704.
[23] T. Xu, C. Wang. “Fuzzy Degree of Image Based on Fuzzy Mathematics”. Journal of computer-aided design and computer graphics, 2002, 14(8): 747 - 749.
[24] X. S. Zhao, S. Z. Chen. “Image Segmentation Based on Global Binarization and Edge Detection ”. Journal of computer-aided design and computer graphics, 2002, 13（2）: 118 - 121.

Adaptively Contrast Enhancement for Image with Genetic Algorithm*

Changjiang Zhang, Xiaodong Wang, Haoran Zhang
College of Mathematics, Physics and Information Engineering, Zhejiang Normal University
Jinhua, Zhejiang Province 321004, China
Email: {zcj74922, wxd, hylt}@zjnu.cn

ABSTRACT

A new contrast enhancement algorithm for image is proposed with genetic algorithm (GA). In-complete Beta transform (IBT) is used to obtain non-linear gray transform curve. Transform parameters are determined by GA to obtain optimal gray transform parameters. In order to avoid the expensive time for traditional contrast enhancement algorithms, which search optimal gray transform parameters in the whole parameters space, a new criterion is proposed. Contrast type for original image is determined employing the new criterion. Parameters space is given respectively according to different contrast types, which shrinks parameters space greatly. The fitness function for GA is formed by two performance measures, namely, contrast measure and noise change measure. Then GA is used to determine the ‘‘optimal’’ set of IBT with the largest fitness function value. Experiment results show that the new algorithm is able to adaptively enhance the contrast of image.

Keywords: Contrast enhancement, In-complete Beta transform, Genetic algorithm

1. INTRODUCTION

Traditional kinds of image enhancement algorithms include: point operators, space operators, transform operators and pseu-color contrast enhancement [1]. Tubbs gives a gray transform algorithm to enhance contrast for images [2]. However, the computation complexity of the algorithm is large. Based on Tubbs's algorithm, Zhou presented a new kind of genetic algorithm to optimize the non-linear transform parameters. Although it can well enhance the contrast for images, its computation requirement is still large. Many existing enhancement algorithms' intelligence and adaptability are bad and much artificial interference is required [3-8]. To solve the above problems, a new algorithm employing IBT and GA is proposed in this paper. The contrast type for an original image is determined by employing a new criterion, where the contrast for original images are classified into seven types: particular dark (PD), medium dark (MD), medium dark slightly (MDS), medium bright slightly (MBS), medium bright (MB), particular bright (PB) and good gray level distribution (GGLD). The IBT operator transforms an original image to a new space. GA is used to determine the optimal non-linear transform parameters. Experimental results demonstrate that the proposed algorithm performs better than the histogram equalization (HE) and un-sharpened mask algorithm (USM).

2. IBT

In Tubbs's algorithm, it uses the unitary incomplete Beta function to approximate the non-linear gray transform parameters [2]. This algorithm searches the optimal parameter α and β in the whole parameter space, where $0<\alpha,\beta<10$. These parameters α and β control the shape of a non-linear transform curve. The incomplete Beta function can be written in the follows [2]:

$$F(u)=B^{-1}(\alpha,\beta)\times\int_0^u t^{\alpha-1}(1-t)^{\beta-1}dt \qquad (1)$$

All the gray levels of an original image have to be unitary before implementing IBT. All the gray levels of the enhanced image have to be inverse-unitary after implementing IBT.

3. CONTRAST CLASSIFICATION FOR IMAGE BASED ON HISTOGRAM

Since an original image has 255 gray levels, the whole gray level space is divided into six sub-spaces: A_1, A_2, A_3, A_4, A_5, A_6, where A_i (i=1, 2, …, 6) is the number of all pixels which distribute in the ith sub-space. Let,

$$M=\max_{i=1}^{6}A_i,\ B_1=\sum_{k=2}^{6}A_i,$$

$$B_2=\sum_{k=2}^{5}A_i,\quad B_3=\sum_{k=1}^{5}A_i,$$

$$B_4=A_1+A_6,\ B_5=A_2+A_3,\ B_6=A_4+A_5,$$

The following classification criterion can be obtained:

$if\ (M=A_1)\ \&\ (A_1>B_1)$

Image is PB;

$elseif\ (B_2>B_4)\ \&\ (B_5>B_6)\ \&$
$(B_5>A_1)\ \&\ (B_5>A_6)\ \&\ (A_2>A_3)$

Image is MD;

$elseif\ (B_2>B_4)\ \&\ (B_5>B_6)\ \&$
$(B_5>A_1)\ \&\ (B_5>A_6)\ \&\ (A_2<A_3)$

Image is MDS;

$elseif\ (B_2>B_4)\ \&\ (B_5<B_6)$
$\&\ (A_1<B_6)\ \&\ (A_6<B_6)\ \&\ (A_4>A_5)$

Image is MBS;

$elseif\ (B_2>B_4)\ \&\ (B_5<B_6)\ \&$
$(A_1<B_6)\ \&\ (A_6<B_6)\ \&\ (A_4<A_5)$

Image is MB;

* This paper is supported by Zhejiang Province Educational Office Foundation (No. 20050292)

elseif $(M = A_6)$ & $(A_6 > B_3)$

Image is PB;

else

Image is GGLD;

end

Where symbol & stands for logic "and" operator.

4. PARAMETERS OPTIMIZATION BY GA

We will employ the GA to optimize transform parameters [9]. If the algorithm is used directly to enhance the global contrast for an image, it will result in large computation burden. The range of α and β can be determined by Tab. I so as to solve above problems.

TABLE 1. Range of α and β

Parameter	PD	MD	MDS	MBS	MB	PB
α	[0, 2]	[0, 2]	[0, 2]	[1, 3]	[1, 4]	[7, 9]
β	[7, 9]	[1, 4]	[1, 3]	[0, 2]	[0, 2]	[0, 2]

Let $\mathbf{x} = (\alpha, \beta)$, $F(\mathbf{x})$ is the fitness function for GA, where $a_i < \alpha, \beta < b_i$（$i = 1,2$）, a_i and b_i（$i = 1,2$）can be determined by Tab.1.

Before running GA, several issues must be considered as follows.

A. System parameters. In this study, the population size is set to 20 and the initial population will contain 20 chromosomes (binary bit strings), which are randomly selected. The maximum number of iterations (generations) of GA is set as 100 respectively in our experiment.

B. Fitness function. In this paper, Shannon's entropy function is used to quantify the gray-level histogram complexity [10]. Given a probability distribution $P = (p_1, p_2, \ldots, p_n)$ with $p_i \geq 0$ for $i = 1, 2, \ldots, n$, and $\sum_{i=1}^{n} p_i = 1$, the entropy of P is:

$$E_{ntr} = -\sum_{i=1}^{n} (p_i \cdot \log p_i) \tag{2}$$

Where $p_i \cdot \log p_i = 0$ by definition for $p_i = 0$. Since P is a probability distribution, the histogram should be normalized before applying the entropy function. Considering noise enlarging problem during enhancement, peak signal-to-noise (psnr) is used to quantify the quality of an enhanced image:

$$P_{snr} = 10 \cdot \log \left(\frac{MN \cdot \max\left(F_{ij}^{\,2}\right)}{\sum_{i=1}^{M}\sum_{j=1}^{N} \left(F_{ij} - G_{ij}\right)^2} \right) \tag{3}$$

Where F_{ij} and G_{ij} are gray level value at (i, j) in original image and enhanced image respectively. M and N are width and height of the original image respectively.

The fitness function is used to evaluate the goodness of a chromosome (solution). In this study, the fitness function is formed by Equation (4):

$$F_{ctr} = -E_{ntr} \cdot P_{snr} \tag{4}$$

Less F_{ctr} is, better contrast of enhanced image is.

C. Genetic operations. In this study, a multi-point crossover is employed. Mutation is carried out by performing the bit inversion operation on some randomly selected positions of the parent bit strings and the probability of applying mutation, P_m, is set to 0.001.

The stopping criterion is the number of iterations is larger than another threshold. Then the chromosome with the smallest fitness function value, i.e., the optimal set of IBT for the input degraded image, is determined. Using the optimal set of IBT enhances the degraded image. Totally there are two parameters in the set of IBT (α and β). The two parameters will form a solution space for finding the optimal set of IBT for image enhancement. Applying GA, the total two parameters will form a chromosome (solution) represented as a binary bit string, in which each parameter is described by 20 bits. We will employ the GA to optimize continuous variables [5].

5. EXPERIMENTAL RESULTS

Fig.1 is relationship curve between number of evolution generation and Best, where $Best = F_{ctr} = -E_{ntr} \cdot P_{snr}$. Fig.2 is a plane image. Fig.3 represents transform curve obtained by GA, where $\alpha = 1.9999$, $\beta = 7.0014$.

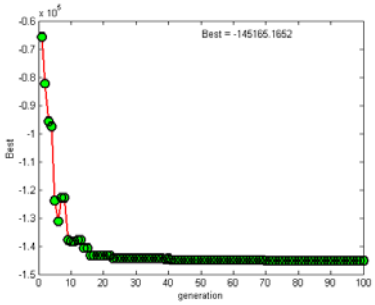


Fig. 1. Evolution curve

Fig.2. Plane image

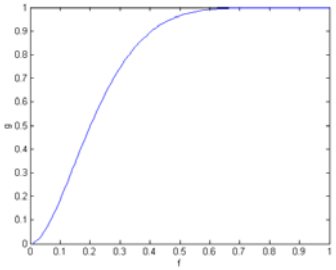


Fig. 3. IBT curve

Fig. 4. Noisy image

Fig.3 is used to enhanced the contrast in Fig.4. Fig.4 is a noisy image, which is added white noise (standard variance of noise is 6.3033). Two traditional contrast enhancement algorithms (HE and USM) are compared with the new algorithm. Fig.5 (a)-(c) are enhanced images by using HE, USM and the new algorithm (NA) respectively.

(a) HE

(b) USM

(c) NA

Fig. 5. Enhancement by three methods

Although the global contrast is good when HE is used to enhance Fig.4, background clutter and noise is also enlarged while some detail information also lost. Although local detail is well enhanced when USM is used to enhance Fig.4, the global contrast is bad and noise is greatly enlarged in Fig.5 (b). The new algorithm can well enhance the contrast in Fig.5 (c), and the background clutter and noise is not been largely amplified. It is very obvious that the new algorithm is better in visual quality than HE and USM.

In order to show the efficiency of the new algorithm, Equation (4) is used to evaluate the enhanced image quality. The fitness function values of HE, USM and NA are 1.8433e+006, 1.2498e+006 and -1.4517e+005 respectively. This can draw the same conclusion as in Fig.5.

6. CONCLUSION

Experimental results show that the new algorithm can adaptively enhance the contrast for an image while keeping the noise in an image from being greatly enlarged. The new algorithm is better than HE and USM in visual quality.

REFERENCES

[1] A. Rosenfield, C.K. Avinash, And Digital Picture Processing, New York: Academic Press, 1982.

[2] J.D. Tubbs, “A note on parametric image enhancement,” Pattern Recognition, 1997, 36(6): 616-621.

[3] S.M. Zhou, Q. Gan, “A new fuzzy relaxation algorithm for image contrast enhancement”, Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, vol.1, 2003, 11-16.

[4] M. Tang, S.D. Ma, et al, “Model-based adaptive enhancement of far infrared image sequences”, Pattern recognition letters, vol.21, 2000, 827-835.

[5] J.A. Stark, “Adaptive image contrast enhancement using generalizations of histogram equalization”, IEEE Transactions on Image Processing, 2000, 9(5): 889-896.

[6] J.Y. Kim, L.S. Kim, et al, “An advanced contrast enhancement using partially overlapped sub-block histogram equalization,” IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11(4): 475-484.

[7] S.J. Yang, J.O. Hwan, et al, “Contrast enhancement using histogram equalization with bin underflow and overflow”, International Conference on image processing, vol.1, 2003, 881 -884.

[8] S.D. Chen, A.R. Ramli, “Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation”, IEEE Transactions on Consumer Electronics, 2003, 49(4): 1301-1309.

[9] M.S. Shyu, J.J. Leou, “A genetic algorithm approach to color image enhancement”, Pattern Recognition, 1998, 31(7): 871-880.

[10] S.J. Yang, Y.J. Park, “Contrast enhancement using histogram equalization with bin underflow and overflow,” International Conference on image processing, vol.1, 2003, 881 -884.

Real-time Motional Image Processing in Intelligent Vehicle System

Ying Yang, Guangyao Zhao
School of Mechanical Engineering and Automation, Northeastern University
Shenyang, Liaoning, 110004, China
Email: yangyingsy@163.com

ABSTRACT

Discusses an algorithm of moving object detection on the basis of algorithmic feasibility and demonstrates how to realize real-time motional object detection. The algorithm detects the changing or moving images by difference method with the image background rebuilt, then combining the continual characteristic of a motional figure array timely and contentedly, the algorithm sets up the historical records of motional foreground image. Processing further the images, the motional direction and speed of objects are calculated to realize the marking and segmented track of motional region. In addition, this algorithm is also suitable for the multiple occurring sometimes. It lays a sound foundation for image follow-up processing and pattern-recognition in video technology with high robustness.

Keywords: Vehicle, Motional Image Processing, Object Detection, Algorithm, Background Rebuilt.

1. INTRODUCTION

Since the information in the traffic environment is mostly from vision. Machine vision technology is an essential component of the intellectual vehicle system. The image processing technology is very important. An intellectual vehicle system, with real-time detector to find the motional goal in the surrounding environment, send out collide alarm, shelter from the barrier, even realize navigation automatically, keeps automobiles running safely and smoothly. Motional images are sampled from the motional targets in different time. The analysis of the motional images is to deal with the figure array so as to study the order of the moving target [1]. In the intellectual vehicle system, it is an indispensable function that the motional goals are detected and tracked. In the detection procedure, the process methods mainly divides into two big classes: Install an array of cameras around the intellectual vehicle, utilize the video image processing method to get the surrounding environment in real time [2,3,4]; Utilize radar technology to obtain the position and speed of the motional goals directly. Radar technology has some advantages in algorithm. But its vision range is small, it can't obtain detailed message of motional goals (such as the automobile type, license plate number, etc.). Therefore although radar technology is still taken as the core in the research, it is combined with the video image processing method in recent years.

2. MOTION DETECTION

During the process of analyzing motional image, we must consider to divide the image into the moving and static areas, and then analyze the changes in the continuous images to distill motional objects.

As the speed of objects changing, the displacement vector of the shape center $(\Delta x, \Delta y)$ is crescent. Area 1 and area 2 are crescent (the maximum is the actual object district). Area 3 reduces gradually, until it becomes zero. The difference between the union set of area 1, 2 and 3 with the actual area increases gradually. The goal overlaps more and more obviously. The most serious situation is when it becomes two times the actual object areas.

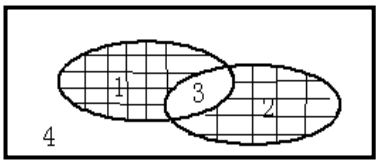


Fig. 1. Difference theory chart

If we know the ideal background $B(x,y)$, and then change the difference between the adjacent frame to the difference between each frame and the background frame [5], obviously no matter how high the speed of movement is, it is impossible to have the objects overlapping phenomenon. So, the key for obtaining the accurate motional object areas are to rebuild the ideal background by the difference definition, as Eq. (1).

$$D_{k,k+1}(x,y) = \begin{cases} 1 & \left| f_k(x,y) - f_{k+1}(x,y) \right| < \theta_0 \\ 0 & \text{other} \end{cases} \quad (1)$$

Among them θ_0 is the two-valued finitude.

Utilize rational restraint criterion to estimate the common background $b_{s,s+1}(x,y)$ of S and $s+1$ frames, the common background $b_{s+1,s+2}(x,y)$ of $s+1$ and $s+2$ frames…, the common background $b_{N-1,N}(x,y)$ of $N-1$ frames and N frames. Then we can utilize the following formula to rebuild the ideal background $b(x,y)$:

$$b(x,y) = \bigcup_{s=0}^{N-1} b_{s,s+1}(x,y) \quad (2)$$

The "$\bigcup$" means: join the common background frames to the estimated background $b(x,y)$ together.

Then the sub-image $Bk(s,j)$ can be judged as the common background of the $s, s+1$ frames, otherwise, it is the sub-image of the motional object area of the $s, s+1$ frames. Obviously, this method is generalized to the multitude motional object condition. So the common background of the $s, s+1$ frames is:

$$b_{s,s+1}(x,y)=\{Bk(s,j)|\ C(Bk(s,j)<TB, j=0,1,\cdots,M-1)\} \quad (3)$$

If realize the rebuilding without any distortion according to the image array $\{f_s(x,y)\}_{s=0}^{N}$, the following formula should be satisfied:

$$b(x,y)\geq B(x,y) \quad (4)$$

The physical meaning of Eq. (3) is that in the random sub-image the motional objects do not exist once at least in the image array $\{f_s(x,y)\}_{s=0}^{N}$. This is the sufficient and essential condition of non-distorting rebuilding. If the length of the frame's arrange is so short that it makes the j sub-image among each $f_s(x,y)$ frame not satisfy with restraint Eq. (4), then choose the very sub-image of the "j" frame that makes this formula reach minimum as the "j" sub-image for background rebuilding [6], that is to definite the frame $\hat{s}$ as following:

$$\hat{s}=\arg\min\{Cons[Bk(s,j)], s=0,1,\cdots,N\} \quad (5)$$

The sub-image $Bk(\hat{s},j)$ in the formula was taken as the "j" sub-image for background rebuilding. When the object is moving in high speed, area 3 in fig. 1. is approximate to zero and area 1 and 2 are all derived from the difference of the background and the motional object, so Eq. (5) means to take the sub-image from all frames that different from the background. Without a doubt, this principle is in accord with people's vision characteristic. The motional area by rebuilding the background is represented as Eq. (6).

$$D_{k,k+1}(x,y)=\begin{cases}1 & If\ |f_k(x,y)-B(x,y)|<\theta_0\\ 0 & Others\end{cases} \quad (6)$$

Fig. 2. The image after differential process

The difference images are obtained on the basis of this method.

3. MOTION SEGMENTATION

According to the principle of relative motion, in the vision system of an intelligent vehicle system, if image gathering device does not move and the change of the array content of the image is caused by external motion, then how to measure the dynamic goal and calculate the motional direction? The algorithm is that inputting and outputting image arrays are all carrying on under the grey level mode that means the image has only one color channels.

When the object is moving, the grey level of the recently received foreground image is set as maximum in the historical record picture of the motion, correspondingly, setting the value of grey level of prospect image at first as minimum, the grey level value of the middle image increases with time pass on, setting up one threshold value in order to segment image better, setting the part which lower than threshold value as zero namely background.

3.1 Update motional historical record

Usually the grey level grade of the grey image is 2^8=256, every image element has 8 bit that equals to one byte, considering the grey level range of the motion template notes is large, so it is set as floating point count. The procedure begins to carry out, the clock calculates time and the historical template begins to note down historical motion. It only notes down the foreground image of recent frames in one second or several seconds, eliminating the image exceeds stipulated time to update the historical backgrounds [7].

3.2 Seek the gradient graph of motion

Process the historical recorded image that is got from the 3×3 *Sobel* operator in the direction of X axle and Y axle, supposing the result that is treated in the place of image element in the direction of X axle and Y axle is $S_x(x,y)$ and $S_y(x,y)$ correspondingly, the gradient direction can be calculated by the following formula

$$A(x,y)=\arctan\left(S_x(x,y)\Big/S_y(x,y)\right) \quad (7)$$

Module of the gradient:

$$M(x,y)=\sqrt{S_x(x,y)^2+S_y(x,y)^2} \quad (8)$$

Fig. 3. Motion gradient graph

The Fig. 3. is the sketch map of direction (angle) by dealing with the historical recorded images. The final value of the motional direction of this object is got by reference of the greatest gradient value.

3.3 Motion segmentation

The purpose of motion segmentation is to divide the foreground motional image in the historical recorded image into different tracking goals, and mark this region as ROI (region of interest) [8]. By setting a threshold value to neglect the very small area, because the too small objects in motional area are very far correspondingly

Fig. 4. Motion segmentation impression chart

Mark all motional regions that in accord with certain

condition, then the detection of motional objects is realized. The motional objects can also be tracked towards different direction in corresponding scene. The method improves the robustness of motion detection. The segmentation result is shown as fig. 4..

4. THE TRACKING OF MOTIONAL OBJECTS

The objects tracking of digital video image is realized by tracking the object in the digital video images array. For this reason, search, detection, segmentation and recognition must be finished. The tracking of digital video image objects is to search and recognize objects in real-time and to ascertain their position and trace. This is a question of searching, recognizing and point describing of random multidimensional signal. To analyze this question, we need to integrate multitude kinds of technological knowledge. The following figure describes the basic task of the tracking system, namely finding objects and marking their positions.

Tracking objects can be found by analyzing motional object's long-term similarity and short-term similarity [9]. After going through long interval, such as (t_l, t_k), the recognition characteristic parameter S_T of the objects has changed greatly, but it is very small in short interval Δt, the recognition characteristic parameter of the objects are stationary in short-term, namely.

$$S_T^k \backsim S_T^{k-1}, t_k - t_{k-1} = \Delta \min \tag{9}$$

In Eq.(9), "$\backsim$" is a similarity sign or comparison operator; S_T^k is marked as $S_T(t_k)$. It is the precondition to realize the long time discerning and tracking for objects with the succession short time discerning and tracking. If this precondition cannot be sufficed, then it is impossible to finish the task of realizing the long time discerning and tracking.

Fig. 5. Multitude objects tracking (tracking time 5s)

The fig. 5. shows the effect of the real-time tracking for multitude objects in different time. This algorithm realizes the segmentation and detecting and the tracking of motional objects. The result of this procedure is good and its robustness is strong. This procedure can detect in real time and achieve the prior purpose.

The tracking time was set five seconds. The tracking algorithm reaches ideal effect of tracking for multitude motional objects comparatively and achieves the aim that algorithms are designed.

It shows that this algorithm can track not only single object but also multitude objects, and its robustness is strong.

5. CONCLUSIONS

In the intelligent vehicle version system, a good algorithm is the key for motional objects detection. A simulation program has been developed based on the algorithmic. The results show this method can be applied for motional object real-time detection; it has the character of efficient and fast. It meets the requirement for high-speed motional object detection and image process. It lays a sound foundation for image follow-up processing and pattern-recognition in video technology with high robustness.

6. REFERENCES

[1] H.W. Liang, "Determine dual threshold from double peak value directly", Pattern Recognize and Artificial Intelligence, 2002, 15(2): 253-256.

[2] M. Bertozzi, A. Broggi, "Artificial vision in road vehicles", Proceedings of the IEEE, 2002, 90(7): 1258-1271.

[3] J. Borenstein, Y. Koven, "Real-time obstacle avoidance for fast mobile robots", IEEE Transactions on Systems, Man, and Cybernetics, 1989, 1179-1186.

[4] Z. Katz, J. Asbury, "On-line position recognition for autonomous AGV navigation systems", Journal of Manufacturing System, 1994, 146-152.

[5] A.K. Hrechak, J.A. Mchug, "Automated fingerprint recognition using structure matching", Pattern Recognition, 1987, 20(4): 429-435.

[6] K. Sugihara, "Some location problems for robot navigation using a single camera", Computer Vision Graphics and Image Processing, 1988, 112-129.

[7] K.T. Sutherland, W.B. Thompson, "Inexact navigation", IEEE International Conference on Robotics and Automationm, 1993, 1-7.

[8] V. Kastrinaki, M. Zervakis, "A survey of video processing techniques for traffic applications", Image and Vision Computing, 2003, Vol.21: 359-381.

[9] Mukherjee J. MRF, "Clustering for segmentation of color images", Pattern Recognition, 2002, 23(8): 917-929.

Edge Detection Using Morphological Directional Gradient*

Xiaopeng Wang, Tao Lei, Yingjie Li
School of Electronic and Information Engineering, Lanzhou Jiaotong University
Lanzhou, Gansu, 730070, China
Email: wangxp1969@sina.com

ABSTRACT

Morphological gradient with threshold is a common way for edge detection, however, it is sensitive to noise, and easily results in false edges, a method for edge detection using morphological directional gradient is proposed, which employs both gradient magnitude and direction to detect edges. Morphological gradient magnitude and threshold are used to detect edges including false edges, and gradient direction with its directional difference threshold is served to check and remove noise caused false edges. Experiments show that this method can efficiently suppress false edges with high signal noise ratio and good localization precision. The effect of edge detection is superior to traditional morphological gradient.

Keywords: Edge Detection, Morphology, Directional Gradient, False Edges, Threshold.

1. INTRODUCTION

Morphological gradient [1,2] combined with threshold is a common way for edge detection, however, it is sensitive to noise, and easily produces false edges. Efficient edge detection method should be the minimum number of false edges, the maximum Signal Noise Ratio (SNR), good localization and robust to noise. To attain these goals, many algorithms have been proposed. For example, Canny [3] adopts three optimal criteria to detect edges, but it always produces double edges; Difference of Estimates operator [4] employs pre-filter to smooth noise and morphological gradient to detect edges, though false edges can be eliminated at some degree, edge localization bias may be aroused.

Strictly speaking, edge contains both magnitude and direction information [5], but traditional morphological gradient (TMG) edge detection only employs edge magnitude, such a case often make the selection of gradient magnitude threshold to be difficult. If the threshold is too low, false edges are hard to remove, while the threshold selected is too high, some weak edges may be lost. In order to get good edges using morphological gradient, a method of morphological directional gradient (MDG) for edge detection is proposed, which combines the gradient magnitude and orientation information into one process of edge detection. First morphological gradient magnitude and orientation are both calculated, and then, on the basis of magnitude threshold, edges are extracted. In order to remove the false edge points in the extracted edge image, the inconsistent gradient directional criteria [6] based on the difference of gradient direction is employed to identify and eliminate false edges.

* Supported by "Qing Lan" Talent Engineering Funds by Lanzhou Jiaotong University

2. MORPHOLOGICAL DIRECTIONAL GRADIENT

Morphological directional gradient operator employs directional structure element, a common used one is the line structure element which is determined by length and angle parameters. Suppose that f is the original image, and $e(l,\alpha)$ denotes line structure element, then MDG operator can be expressed as following.

$$g_\alpha = [f \oplus e(l,\alpha)] - [f \Theta e(l,\alpha)] \quad (1)$$

where g_α is MDG along direction α, $\oplus$ and Θ denote morphological dilation and erosion operator, respectively. It can be seen that g_α contains both magnitude and direction information, from which we can easily estimate the magnitude and direction respectively.

Estimation of edge gradient magnitude

Let $G(l,\alpha)$ denote the gradient magnitude of g_α. In order to simply the computation, α is limited to several finite angle, that is, α belong to $\{0, \pi/4, \pi/2, \pi 3/4\}$, and then horizontal/vertical gradient magnitude M_{HV} can be estimated by following.

$$M_{HV} = \sqrt{G^2(l,0) + G^2(l,\pi/2)} \quad (2)$$

where $G(l,0)$ and $G(l,\pi/2)$ denote the horizontal and vertical gradient magnitude of g_α, respectively.

Similarly, diagonal gradient magnitude M_D is

$$M_D = \frac{1}{\sqrt{2}}\sqrt{G^2(l,\pi/4) + G^2(l,3\pi/4)} \quad (3)$$

where $G(l,\pi/4)$ and $G(l,3\pi/4)$ denote the two diagonal gradient magnitude of g_α, respectively, the constant $1/\sqrt{2}$ is only to meet the need of magnitude unitary.

In order to analyze the principle of gradient magnitude simply, suppose that gradient image $g(x,y)$ from Eq. (1) is expressed as

$$g(x,y) = a + b(x\cos\theta + y\sin\theta) \quad (4)$$

where a is a constant corresponding to the background of $g(x,y)$, b is the gradient edge magnitude, and θ represents the gradient edge direction. Then four directional gradient edge magnitudes can be calculated as following.

$$\begin{cases} G(l,0) = 2b|\cos\theta| \\ G(l,\pi/4) = 2b|\cos\theta + \sin\theta| \\ G(l,\pi/2) = 2b|\sin\theta| \\ G(l,3\pi/4) = 2b|\cos\theta - \sin\theta| \end{cases} \quad (5)$$

According to Eq. (2) and (3), M_{HV} and M_D are respectively obtained as following.

$$M_{HV} = \sqrt{G^2(l,0) + G^2(l,\pi/2)} = 2b \quad (6)$$

$$M_D = \sqrt{G^2(l,\pi/4) + G^2(l,3\pi/4)} = 2b \quad (7)$$

In ideal case, we can see that M_{HV} and M_D are both equal to $2b$. In general, because of noise and other factors, M_{HV} and M_D may be no longer equal, in order to reduce the effect of direction to the estimation of edge gradient magnitude, the gradient magnitude can be estimated by the arithmetic average of M_{HV} and M_D.

$$M_a = (M_{HV} + M_D)/2 \quad (8)$$

Estimation of gradient direction

Like the estimation of the edge gradient magnitude, gradient direction can also be estimated by four gradient magnitude components of $G(l,\alpha)$. Let θ_{HV} and θ_D be the horizontal/vertical and diagonal gradient direction, respectively, then θ_{HV} can be calculated by arc tangent.

$$\theta_{HV} = \arctan\frac{G(l,\pi/2)}{G(l,0)} \quad (9)$$

Since diagonal $G(l,\pi/4)$ and $G(l,3\pi/4)$ can be viewed as rotated $\pi/4$ from $G(l,0)$ and $G(l,\pi/2)$, the calculation of θ_D needs to be rotated back $\pi/4$ as following.

$$\theta_D = \begin{cases} \pi/4 - \arctan\dfrac{G(l,3\pi/4)}{G(l,\pi/4)} & (\theta_D \in [0,\pi/4]) \\ \pi/4 + \arctan\dfrac{G(l,3\pi/4)}{G(l,\pi/4)} & (\theta_D \in [\pi/4,\pi/2]) \end{cases} \quad (10)$$

For the gradient model in Eq. (4), it can be verified that both θ_{HV} and θ_D are equal to θ. However, due to the gradient being non-negative, θ_{HV} and θ_D are both limited to $[0,\pi/2]$, in order to estimate the gradient direction in $[-\pi/2,\pi/2]$ range, we segment $[-\pi/2,\pi/2]$ into four uniform ranges, in each range, different expressions are used to estimate the gradient direction θ_{HV} and θ_D, respectively.

Suppose that $\eta_1 = \dfrac{G(l,\pi/2)}{G(l,0)}$ and $\eta_2 = \dfrac{G(l,3\pi/4)}{G(l,\pi/4)}$, then we can obtain from Eq.(9) and (10) that if $\eta_1 < 1$ then $\theta_{HV} \in [0,\pi/2]$, and $\eta_2 > 1$, then $\theta_{HV} \in [-\pi/2,0]$. The case of θ_D is the same as θ_{HV}. The final estimation of θ_{HV} and θ_D can be rewritten as following.

If $\eta_1 \le 1$ and $\eta_2 \le 1$, θ_{HV} and θ_D belong to $[0,\pi/4]$,

$$\begin{cases} \theta_{HV} = \arctan(\eta_1) \\ \theta_D = \pi/4 - \arctan(\eta_2) \end{cases} \quad (11)$$

if $\eta_1 \ge 1$ and $\eta_2 \le 1$, θ_{HV} and θ_D belong to $[\pi/4,\pi/2]$,

$$\begin{cases} \theta_{HV} = \arctan(\eta_1) \\ \theta_D = \pi/4 + \arctan(\eta_2) \end{cases} \quad (12)$$

if $\eta_1 \le 1$ and $\eta_2 \ge 1$, θ_{HV} and θ_D belong to $[-\pi/4,0]$,

$$\begin{cases} \theta_{HV} = -\arctan(\eta_1) \\ \theta_D = \pi/4 - \arctan(\eta_2) \end{cases} \quad (13)$$

if $\eta_1 \ge 1$ and $\eta_2 \ge 1$, θ_{HV} and θ_D belong to $[-\pi/4,-\pi/2]$,

$$\begin{cases} \theta_{HV} = -\arctan(\eta_1) \\ \theta_D = -(3\pi/4) + \arctan(\eta_2) \end{cases} \quad (14)$$

By applying Eq. (11) ~ (14), we can simply estimate the gradient direction θ_{HV} and θ_D from four gradient components in $[-\pi/2,\pi/2]$ range.

3. FALSE EDGES REMOVAL

Theoretically, edges are only related to their gradient magnitude. When gradient magnitude threshold is solely used to detect edge, higher magnitude parts are fully extracted, however, noise caused false edges may have higher magnitude, therefore, only gradient magnitude threshold is hard to obtain real edges. For the purpose to remove these false edges, we tend to apply gradient direction to remove false edges in the edge image after gradient magnitude threshold. The criterion for checking false edge is the inconsistent gradient direction.

In ideal case, gradient direction θ_{HV} and θ_D should be equal [6]. However, when the image is corrupted by noise, θ_{HV} and θ_D will be inconsistent, such an inconsistency will help us to identify and remove false edges from detected edge image. The absolute value $|\theta_{HV} - \theta_D|$ with directional difference threshold can be taken as the criterion for false edge suppression.

Suppose that $\theta_e = |\theta_{HV} - \theta_D|$, θ_T denotes the gradient directional difference threshold, and M_T is the gradient magnitude threshold. When edge is extracted from background using gradient magnitude threshold M_T, then we use inconsistent gradient direction to check false edges. If $\theta_e < \theta_T$, the current edge pixel will be identified as edge point, otherwise, they may be false edge points and will be removed.

4. ALGORITHM IMPLEMENTATION

Unlike traditional threshold process, proposed method employs two different thresholds to extract potential edges and to remove false edges respectively. The combination of gradient magnitude threshold with direction difference threshold efficiently avoid the false edge appear.

The whole edge detection process can be summarized as following steps.

1) Input original image f and select line structure element $e(l,\alpha)$.

2) Use Eq. (1) to calculate morphological gradient magnitude corresponding to four selected directions, and then to obtain $G(l,0)$, $G(l,\pi/4)$, $G(l,\pi/2)$ and $G(l,3\pi/4)$.

3) Apply Eq. (2) ~ (3) to estimate M_{HV} and M_D, respectively, and employ Eq. (8) to calculate M_a.

4) Estimate gradient direction θ_{HV} and θ_D by Eq. (11) ~ (14), then obtain θ_e, that is, $\theta_e = |\theta_{HV} - \theta_D|$.

5) Detect edge image by selected gradient magnitude threshold M_T.

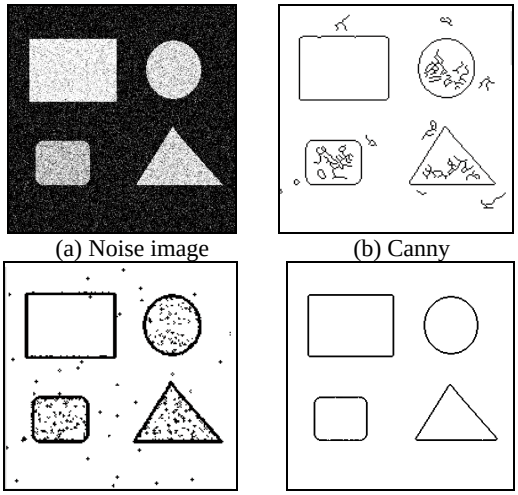
6) For a given edge point, if $M_a \geq M_T$, then apply θ_e to check whether it is the real or the false edge point, if $\theta_e < \theta_T$, it is a real edge point, else, is a false edge point, and throw it out from edge image.

7) Repeatedly perform step 6) till all the edge points are detected from step 5).

8) Output edge image.

5. EXPERIMENTS

Synthetic image with additional 5% Gaussian noise is firstly used as the original (Fig.1 (a)). Fig.1 (b) gives the detected edge image using Canny operator, and Fig.1 (c) and (d) show the edge detection result by using TMG with gradient magnitude threshold (threshold is 8) and proposed method ($l = 3$, $M_T = 6$, $\theta_e = 1.2^0$), respectively. We notice there exist false edges in Fig.1 (b) and (c), it is obvious that the visual effect of proposed method as shown in Fig.1 (d) is superior to Canny and TMG in the case of noise.



(a) Noise image (b) Canny

(c) TMG (d) MDG

Fig. 1. Edge of synthetic noise image using different method. .

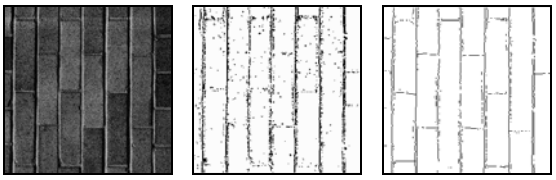
From the evaluation of edge localization, we also obtain from Fig.1 that the average localization precision of MDG can reach 98.3%, Canny and TMG are 96.2% and 94.3%, respectively.

(a) Canny (b) TMG (c) MDG

Fig. 2. Edge of natural image using different method.

Fig. 2 (a) gives the edge of Lena image with noise (additional 5% Gaussian) by using canny operator, Fig. 2 (b) and (c) show the edge images of TMG and proposed MDG methods, respectively. We can see that TMG is more sensitive to noise, and results in more false edges. Though canny edge is superior to TMG method, it still has small part of false edges. Compared with TMG, MDG edge has better performance for suppressing false edges.

In order to compare the visual effect of edge detection of TMG and MDG for directional image, Fig. 3 (a) is selected as the original image, which contains horizontal and vertical directional brick edges. Fig.3 (b) and (c) give the edge images of Fig. 3 (a) by using TMG and MDG methods, respectively. We notice from Fig.3 (b) that some edges are lost and false edges are generated by TMG. Compared with TMG, most false edges are suppressed without more real edges losing by using MDG method. This indicates that for directional image MDG is more efficient than TMG.



(a) Original image (b) TMG (c) MDG

Fig. 3. Edge of natural image using different method.

Table.1 shows the SNR of different methods under various noises by using the following expression.

$$SNR = 10 \lg \frac{|I|^2}{|I - I_N|^2} \quad (15)$$

Where I and I_N denote ideal noiseless edge image and noise edge image, respectively. Comparisons from Table 1. show that MDG has higher SNR than Canny and TMG.

Table 1. SNR of different methods under various noises

Noise type	Gaussian (0.05)	Salt & pepper (0..05)	Poisson	Speckle 0.04
Canny	45.1	42.6	43.56	48.30
TMG	22.2	24.3	30.9	41.6
MDG	67.0	51.3	43.6	57.5

6. CONCLUSIONS

A method for edge detection by morphological directional gradient is proposed, which employs both gradient magnitude and direction to detect edges. By using inconsistent gradient direction criteria, false edges caused by noise are checked and removed. The sensitivity of traditional morphological gradient to noise is greatly improved. Experiments show that this method can not only suppress false edges, but has the higher SNR and good localization precision.

7. REFERENCES

[1] P.Soille, Morphological image analysis: principles and applications, New York: Springer-Verlag, 2003.

[2] S.Wood, G. Qu, “A modified gray level morphological gradient with accurate orientation estimates and reduced noise sensitivity”, 34th IEEE Asilomar Conference on Signal, System and Computers, 2000.

[3] Y.J. Zhang, Image segmentation, Beijing: Science Press, 2001.
[4] J. Yoo, C.A. Bouman, et al, “The Nonlinar prefiltering and difference of Estimates Approaches to Edge detection: Applications of Stack Filters”,Graphical Models and Image Processing,1993,55,(2): 140-150.
[5] A.P. Paplinkski, “Directional Filtering in Edge Detection”,IEEE Trans. Image Processing, 1998,7, (4): 611-615.
[6] G. Qu, “Directional Morphological Gradient Edge Detector”, PhD thesis, Santa Clara University, California, 2001.

Application of Wavelet Multi-scale Product in Image Edge Detection Technology

Xiaoli Zhao
College of Electronic and Electrical Engineering, Shanghai University of Engineering Science
Shanghai 201620, China
Email: evawhy@163.com

ABSTRACT

In order to decrease the influence of noise on edge detection, wavelet multi-scale product is adopted to detect image edge; and improvement has been made to the method of seeking local maximum module. The results of experiment show that wavelet multi-scale product has strong ability of controlling noise and can achieve good effects of edge detection.

Keywords: Wavelet Multi-scale Product, Edge Detection, Local Maximum Module.

1. INTRODUCTION

Edge detection is one of the important aspects in image processing and computer vision [1]. Image edge, being the basic feature of image and involving important information, corresponds to the abrupt or discontinuous position.

Traditional edge detection methods are based on convolution using spatial domain derivative operator such as Roberts operator, Prewitt operator, Sobel operator and Laplacian operator [2],which are implemented through either maximum magnitude of fist-derivative or zero-crossing of second-derivative in original image. Considerable interest has arisen in recent years in Marr operator (LoG operator)[3] and Canny operator [4]. The principle of Canny operator is first to smooth image with Gauss function and second to determine edge with maximum magnitude of fist-derivative; The principle of LoG operator is to make Laplacian computation to convolution of image and Gauss function, then define zero-crossing of second-derivative as edge. Although these algorithms have their own characteristics, contradiction of noise and position precision still exists in practical application.

Wavelet analysis is called "mathematical microscope", which can see not only panorama but also detail of signal. Image multi-scale edge detection algorithm based on wavelet transform can well denoise and establish edge.

2. EDGE DETECTION BASED ON WAVELET MULTI-SCALE PRODUCT

The basic task of edge detection is to resolve contradiction between detection precision and noise. Now some edge detection algorithms [5,6], with selecting proper wavelet basis, achieve wavelet transform of image horizontal and vertical direction in a certain scale, and then compute local maximum module point in terms of the direction of edge. These points are edge points if they meet the threshold condition. In order to well control noise influence on edge detection, wavelet multi-scale product is adopted to distinguish image edge and noise.

2.1 Definition of Wavelet Multi-scale Product

M level wavelet transform is implemented on image *f(x, y)*. Two-dimension multi-scale product of point *(x, y)* in x and y direction is denoted respectively as:

$$pr^1_{2^M}(x,y) = \prod_{j=1}^{M} W^1_{2^j} f(x,y) \quad (1)$$

$$pr^2_{2^M}(x,y) = \prod_{j=1}^{M} W^2_{2^j} f(x,y) \quad (2)$$

According to Lipschitz exponent theory, noise Lipscltitz exponent is $a<0$ and signal Lipscltitz exponent is $a>0$, that is, wavelet transform amplitude of noise decreases with increasing of scale, however wavelet transform amplitude of signal increases or remains constant with increasing of scale[1]. From above conclusion is drawn that the multi-scale product of noise reduces rapidly and that of signal increases rapidly, as is shown in Fig. 2.: wavelet transform noise lies in scale 3, whereas multi-scale product noise is next to 0 and signal magnitude increases rapidly, which means multi-scale product has advantages in controlling noise and extract edge.

2.2 Selection of Wavelet Basis

Selection of wavelet basis also influences detection result in edge detection of wavelet transform. Edge detection using B-spline wavelet has advantages of less computation, higher locating precision because B-spline wavelet is compact support, symmetric.

According to Mallat fast algorithm, the essence of filter group and multi-resolution analysis is same, so decomposition and reconstruction of *f(x)* can be realized with filter group and coefficient, instead of computing wavelet transform. The relation of two-scale is written as:

$$\frac{1}{2}\phi_m(\frac{x}{2}) = \sum_{k=-\infty}^{\infty} h_k \phi_m(x-k) \quad (3)$$

$$\frac{1}{2}\psi_m(\frac{x}{2}) = \sum_{k=-\infty}^{\infty} g_k \phi_m(x-k) \quad (4)$$

Table 1. shows filter value of third B-spline wavelet.

Table 1. Filter value of third B-spline wavelet

k	-2	-1	0	1	2
h_k	0.0625	0.25	0.375	0.25	0.0625
g_k	0	0	−2	2	0

2.3 Processes of Wavelet Multi-scale Product Edge

Detection

1) M level($M \leqslant 3$)wavelet transform is implemented on image $f(x, y)$;

2) Wavelet multi-scale product pr^1 and pr^2 are computed in x and y direction in terms of Eq.(1)and(2);

3) Module $M_{2^j} f(x,y)$ and direction angle $A_{2^j} f(x,y)$ are computed using Eq. (5) and Eq. (6);

$$M_{2^j} f(x,y) = \sqrt{|\, pr^1 \,|^2 + |\, pr^2 \,|^2} \tag{5}$$

$$A_{2^j} f(x,y) = \arctan\left(\frac{W^2_{2^j} f(x,y)}{W^1_{2^j} f(x,y)}\right) \tag{6}$$

4) Local maximum module is sought along direction angle, achieving image of maximum module (for detailed processes, see Improved Method of Seeking Local Maximum Module)

5) Binary image is achieved by filtering false edge;

2.4 Improved Method of Seeking Local Maximum Module

A method of local maximum module is presented in [7]. Algorithm is improved on the basis of researching.

If $M_{2^j} f(x,y)$ is local maximum module of point (x, y)in gradient direction $A_{2^j} f(x,y)$, 8-neighbor of point(x, y) has 4 condition shown in Fig. 1., namely, $M_{2^j} f(x,y)$ is only wavelet transform local maximum module of three pixel in one direction which is horizontal,vertical,35^0,135^0 direction. Because main value range of inverse tangent is $(-\frac{\pi}{2}, \frac{\pi}{2})$, direction angle $A_{2^j} f(x,y) \in (-\frac{\pi}{2}, \frac{\pi}{2})$, so improved algorithm is as follows:

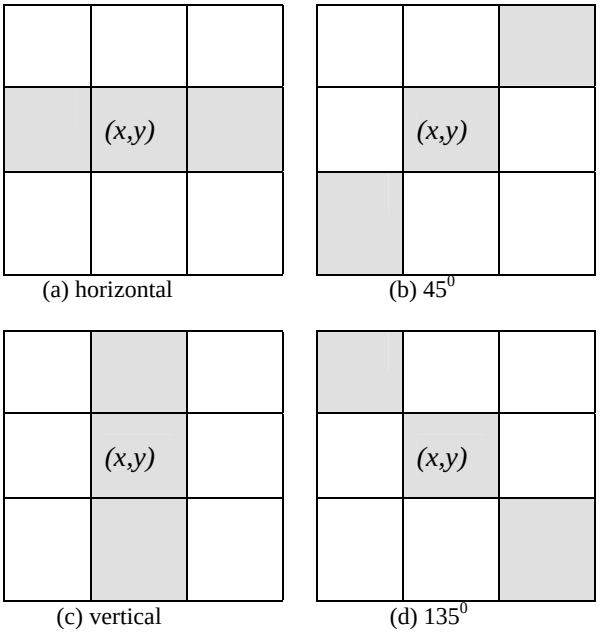
1) if $|\, A_{2^j} f(x,y) \,| \leqslant \frac{\pi}{8}$,comparison is made among points(x,y-1), (x, y), (x,y+1), then proceed to 5);

2) if $A_{2^j} f(x,y) > \frac{\pi}{8}$ and $A_{2^j} f(x,y) \leqslant \frac{3\pi}{8}$,comparison is made among points(x+1,y−1), (x,y), (x−1,y+1), proceed to 5);

3) if $A_{2^j} f(x,y) < -\frac{\pi}{8}$ and $A_{2^j} f(x,y) \geqslant -\frac{3\pi}{8}$,comparison is made among points(x−1,y−1), (x,y), (x+1,y+1), proceed to 5);

4) if($|\, A_{2^j} f(x,y) \,| > \frac{3\pi}{8}$)and ($|\, A_{2^j} f(x,y) \,| < \frac{\pi}{2}$ = or($W^1_{2^j} f(x,y) = 0$),comparison is made among points(x−1,y), (x,y), (x+1,y), proceed to 5);

5) If $M_{2^j} f(x,y)$ is at or above magnitude of other two points and strictly above one of other two points, $M_{2^j} f(x,y)$ is local maximum module, otherwise $M_{2^j} f(x,y)$ is 0.



(a) horizontal (b) 45^0 (c) vertical (d) 135^0

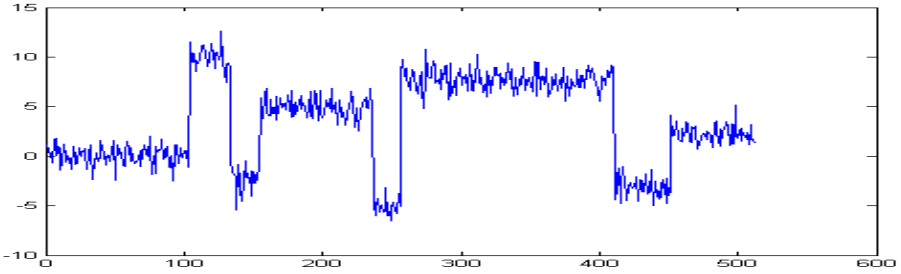
Fig. 1. 4 condition of 8-neighbor

3. EXPERIMENT RESULTS

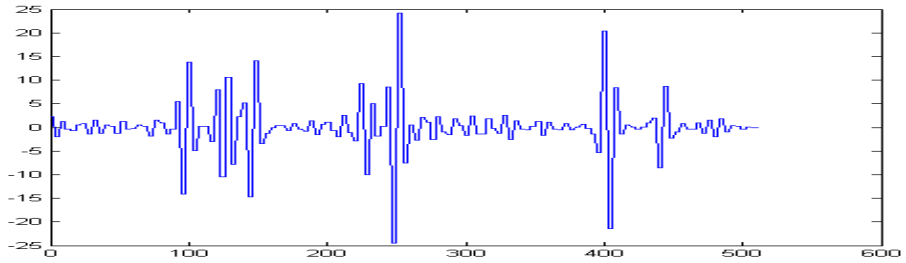
In Fig. 3. (a) is original image with noise. It is processed with method of multi-scale product in scale 3 and improved maximum local module, and Fig. 3. (b) is result of detection. The result shows that this method has strong ability of controlling noise. Achieved edge, however, needs further edge linking.

4. CONCLUSION

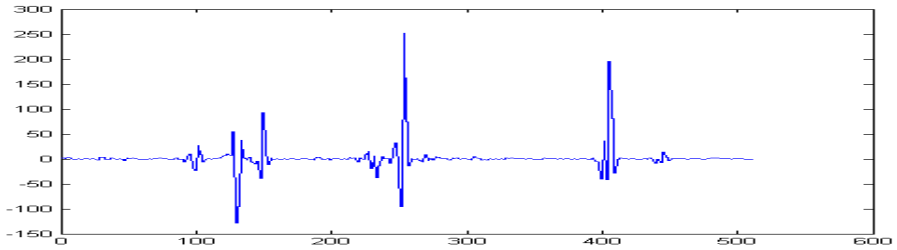
The basic task of edge detection is to resolve contradiction between detection precision and noise. In this paper wavelet multi-scale product is adopted to detect image edge and the method of seeking local maximum module is improved. The experiment result shows that this method has strong ability of controlling noise and can achieve good effect of edge detection. Because edge of image, in some sense, has some width, in order to obtain better edge image and locate precise edge, further research is edge linking and sub-pixel location.



(a) Original image with noise



(b)Wavelet transform in scale 3



(c)Multi-scale product in scale 3

Fig. 2. Signals of wavelet transform

(a) image with noise

(b) result of edge detection

Fig. 3. Result of experiment

5. REFERENCES

[1] X.H. Ding, Research on Vision Inspection Based on Wavelet Analysis. Unpublished Dissertation, Hefei University of Technology, Anhui, 2003.

[2] R.Kenneth, Castleman, Digital Image Processing. Tsinghua University Press, Beijing, 2003.

[3] D. C Marr, E.C. Hildreth, Theory of Edge Detection. Proc Roy Soc, London, 1980.

[4] Canny, J. A, Computational Approach to Edge Detection. IEEE Trans Pattern Anal Machine Intelligence, 1986, 8(6): 679-698.

[5] Mallat, S, W L Hwang. Singularity Detection and Processing with Wavelets. IEEE Trans. Information Theory.1992, 38(2): 617-643.

[6] G. R. Gao, A Revised Image Edge Detection Method Based on Wavelet. Wuhan University. (Nat. Sci. ed.), 2005, 51(5): 615-619.

From molecule to man: the system science of decision support in individualized e-Health

Peter M.A. Sloot
Section Computational Science, Peter M.A. Sloot, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
Email: P.M.A.Sloot@uva.nl

ABSTRACT

Distributed computing technology is used to build an integrated decision support system for patient specific medication. It combines simulation of the disease and the drugs with advanced data processing.

Keywords: Grid Computing, HIV simulation, Decision Support.

1. FROM MOLECULE TO MAN

'During the next decade, the practice of medicine will change dramatically, through personalized, targeted treatments that will enable a move beyond prevention to pre-emptive strategies.' [1]

We are a complex system: from a biological cell, made of thousands of different molecules that work together, to billions of cells that build our tissue, organs and immune system, to our society, six billion unique interacting individuals. Such complex systems are not made of identical and undistinguishable components: rather each gene in a cell, each cell in the immune system, and each individual have their own characteristic behavior and provide unique value and contributions to the systems in which they are constituents. The complete cascade from the genome, proteome, metabolome, physiome to health constitutes multi-scale, multi-science systems, and crosses many orders of magnitude in temporal and spatial scales [2] (see *Fig. 1*). The interactions between these systems form exquisite multi-tiered networks, each component being in non-linear contact with many selected interaction partners. These networks are not just complicated; they are complex. Understanding, quantifying *and* handling this complexity is one of the biggest scientific challenges of our time [3].

It is the central assertion of this presentation is that computer science is *the* language to study and understand these systems, and that the same laws and organizing principles that dictate biomedical systems are reflected in the architecture of simulating computer systems.

We discuss some of these laws and organizing principles required to build systems for *individualized* biomedicine that can account for variations in physiology, treatment and drug response.

1.1 Pushing and pulling

We observe an *application pull* from biomedicine with the change in paradigm to *in silico* studies, where more and more details of biomedical processes are simulated in addition to *in vivo* and *in vitro* studies. These simulated processes are being used to support medical doctors in making decisions through exploration of different scenarios. Typical examples are pre-operative simulation and visualization of vascular surgery[4], and expert systems for drug ranking[5]. At the same time we observe a *technology push* from computing and data availability[6].

In the field of high-performance computing there have been changes from sequential to parallel to distributed computing, where the 'killer applications' moved from mathematics to physics to chemistry to biology to medicine, thus increasing the complexity of the systems under study with the complexity of the computational systems. In addition, with the advances in Internet technology and Grid computing [7], huge amounts of data from sensors, experiments and simulations have become available. There are, however, significant computational, integration, collaboration, and interaction gaps between these observed application pull and technology push.

1.2 Bridging the gaps

In order to close the *computational gap* in systems biology, we need to construct, integrate and manage a plethora of models. A bottom-up data-driven approach will not work for this. Web and Grid services are needed to integrate often incompatible applications and tools for data acquisition, registration, storage, provenance, organization, analysis and presentation, thus bridging the *integration gap*. Even if we manage to solve the computational and integration challenges, we still need a system-level approach to share processes, data, information and knowledge across geographic and organizational boundaries within the context of distributed, multidisciplinary and multi-organizational collaborative teams, or 'virtual organizations' as they are often called, thus closing the *collaboration and interaction* gap.

Finally, we need intuitive methods to streamline all these processes dynamically depending on their availability, reliability and the specific interests of the end-users (medical doctors, surgeons, clinical experts, researchers). Such methods can be captured into 'scientific workflows' in which the flow of data and action from one step to another is expressed in a workflow language [8,9]. A general scheme for conducting such e-Science research is depicted in Fig. .

In this lecture we will discuss the development of a Grid based decision support system consisting of modules such as the one depicted in Fig. , for individualized drug ranking in Human Immunodeficiency Virus (HIV-1) diseases, called *ViroLab* [10]. The reason for using this complex problem of HIV drug resistance as a prototype for our system-level approach is twofold. First of all, HIV drug resistance is becoming an increasing problem worldwide, with a considerable number of HIV infected patients developing failure of complete suppression of the virus despite combination therapy with antiretroviral

drugs. Second, HIV drug resistance is one of the few areas in medicine where genetic information is widely available and used for a considerable number of years. As a consequence, large numbers of complex genetic sequences are available, in addition to clinical data. In addition we will discuss some new results on modeling and in particular Prof. C. Boucher and Dr. D. van de Vijver from the University Medical Centre Utrecht, The Netherlands. This research was supported by the Dutch Bsik project VLe: Virtual Laboratory for e-Science and the European *ViroLab* grant INFSO-IST-027446.

3. REFERENCES

[1] W.H. Frist, "Health Care in the 21st Century", New England Journal of Medicine, Vol. 352, Jan. 2005, pp. 267-272.

[2] A. Finkelstein, J. Hetherington, L. Li, O. Margoninski, P. Saffrey, R. Seymour and A. Warner, "Computational Challenges of System Biology", Computer, Vol. 37, No. 5, 2004, pp. 26-33.

[3] A. Barabasi, "Taming Complexity", *Nature Physics*, Vol 1, November 2005, pp 68-70.

[4] A. Tirado-Ramos, P.M.A. Sloot, A.G. Hoekstra and M. Bubak, "An Integrative Approach to High-Performance Biomedical Problem Solving Environments on the Grid", *Parallel Computing*, Special Issue on High-Performance Parallel Bio-computing, Vol. 30, nr 9-10, 2004, pp. 1037-1055.

[5] P.M.A. Sloot, A.V. Boukhanovsky, W. Keulen, A. Tirado-Ramos and C.A. Boucher, "A Grid-Based HIV Expert System", Journal of Clinical Monitoring and Computing, Vol. 19, Numbers 4-5. October 2005 pp. 263 – 278.

[6] A.J.G. Hey and A.E. Trefethen, "The Data Deluge: An e-Science Perspective", Grid Computing - Making the Global Infrastructure a Reality, chapter 36, 2003, pp. 809-824.

[7] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations", International Journal of High Performance Computing Applications, Vol. 15, 2001, pp. 200-222.

[8] l. Altintas, A. Birnbaum, K.K. Baldridge, W. Sudholt, M. Miller, C. Amoreira, Y. Potier and B. Ludäscher, "A Framework for the Design and Reuse of Grid Workflows", SAG 2004, Beijing, China, September 20-24, 2004, Vol. 3458 (3), pp. 119-132.

[9] F. Neubauer, A. Hoheisel and J. Geiler, "Workflow-based Grid applications", Future Generation Computer Systems, Vol. 22, Issues 1-2, January 2006, pp. 6-15.

[10] ViroLab, EU project INFSO-IST-027446, http://www.virolab.org/.

4. ACKNOWLEDGES

The author would like to thank the *ViroLab* consortium

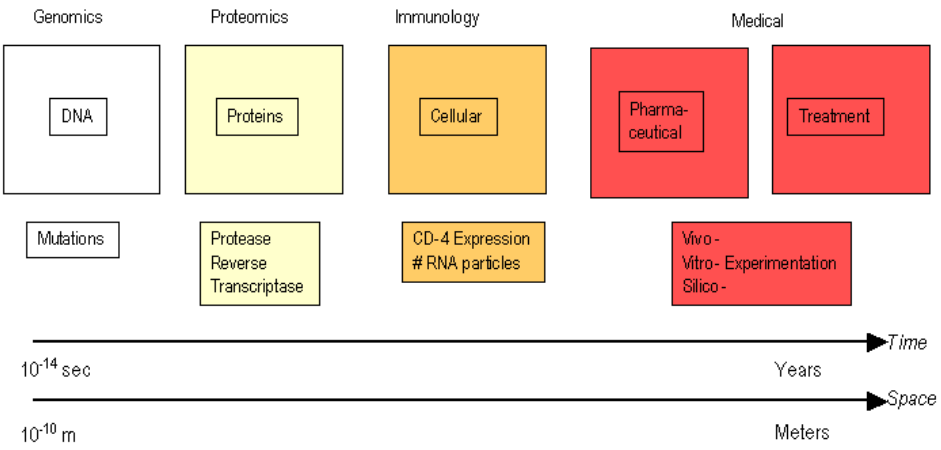


Fig. 1. Multi-scale, multi-science models and techniques are needed to cover the huge spatial and temporal scales in studying drug response in infectious diseases

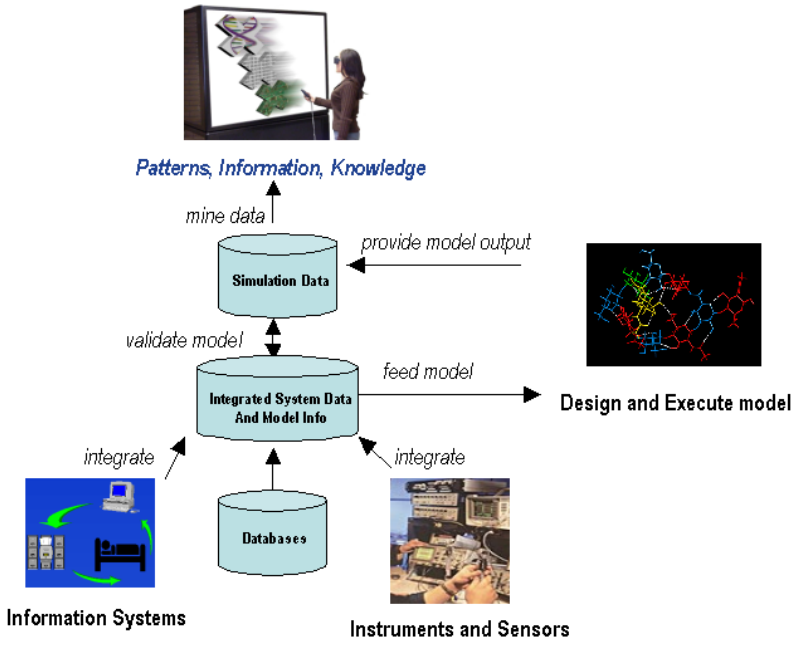


Fig. 2. General architecture for conducting e-Science research: information systems integrate available data with data from specialized instruments and sensors into distributed repositories. Computational models are then executed using the integrated data, providing large quantities of model output data, which is mined and processed in order to extract useful knowledge.

Evolutionary Algorithm Based on Chaos Mutation and Immune Selection*

Nan Zhang
School of Computer Science and Technology, Southwest University for Nationalities
Chengdu, 610041, China
Email: panda_zn@163.com
And
Jianhua Zhang and Zhishu Li
School of Computer, Sichuan University
Chengdu, 610065, China
Email: lzshu@scu.edu.cn
And
Xianze Yang, Bing Wu and Chunchang Fu
School of Computer Science and Technology, Southwest University for Nationalities
Chengdu, 610041 , China
Email: yangxianze@swun.cn

ABSTRACT

In order to keep the population's diversity, avoid local optimization and improve performance of evolutionary algorithm, a chaos immune evolutionary algorithm is presented. The over-spread character of chaos sequence was used to overcome redundancies, and the chaos initial value sensitivity was used to enlarge the searching space. At the same time, an immune selection operator was used to adjust the density, and vaccinations were used to exert the advantage of individuals. The experimental results show that the algorithm has good performance in the aspects of both convergence speed and the global convergence.

Keywords: Immune, Chaos, Evolutionary Algorithm, and Initial Value Sensitivity.

1. INTRODUCTION

Evolutionary Algorithm (EA)[1] is a new calculating method, which follows evolution and inheritance process in biology, develops high quality solutions, washes out low quality ones, advances whole quality of population step by step, and finally approaches the optimal solution. Because it is easy to run into local convergence in application, the algorithm is restricted in practice. Forrest and her fellows put forward an immune system [2], in which pattern recognition and study can be done in individual level and population level. In this mode, an antibody set was evolved after a certain antigen set was identified by it. Afterwards, many scholars also presented some algorithms [3] binding AIS and EA to keep population diversity and eliminate the phenomena of juvenility constringency. Those algorithms were already applied in many fields.

Though Immune Evolutionary Algorithm can limit solutions in a small space, it is not still satisfied with the efficiency of searching to get the optimal solution. Considering immune systems have characters of chaos and the mutation in evolvement can be thought as a chaos dynamics process, we present a Chaos Immune Evolutionary Algorithm to improve the algorithm performance in this text.

* This work is partially supported by the National Hi-Tech R&D Program (863) of China Grant #2002AA144020 to Zhishu Li.

2. THE CHAOTIC CHARACTER OF IMMUNE SYSTEM

Chaos [4] is the circulative action with unfixed cycle, generated by some dynamical systems. Chaos is the result of interaction between the character of dissipation and nonlinearity, which is applied in some fields widely, including engineering, physics, mathematics and biology. The effect of the dissipation makes the chaos system behave with stabilization and contraction of the volume in the whole and the great scope. At the same time, the effect of nonlinear function makes the system track be unstable partially and the unstable character causes the track to separate partially. The stable character of the whole and the unstable character of the part make system display a complex movement format, and form the strange behavior of chaos. The essential character of chaos is the initial value sensitivity, that is to say, the mini difference of the initial state would bring the great difference of the result.

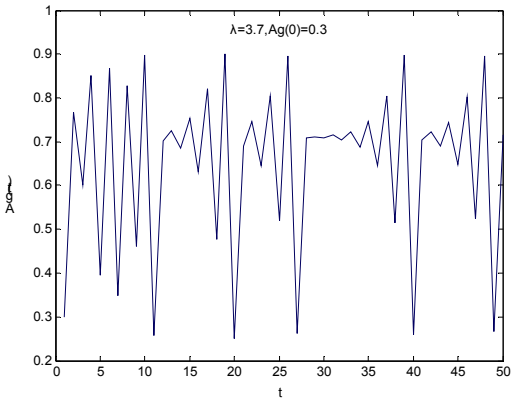
Immune system is a complex system, which was formed by the different levels, and each level contains many kinds of antigen cells. The non-linear mathematical instrument can be used to describe the different antigen cell mutual function. From the time evolutionary process of various antigen cells we can understand the movement rule, mechanism and function of immune system [5].

After the change manner of antigen cell's density was looked upon in immune system, the multiplication of antigen can be represented by (1):

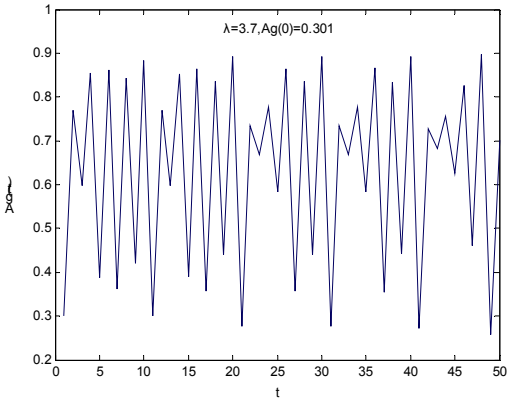
$$Ag(t+1) = k(1-\frac{Ag(t)}{Ag_{\max}})Ag(t) - k'Ag(t) \qquad (1)$$

Where, *Ag (t)* is the density of antigen at time *t*, and the density at time *t+1* can be calculated from the status at time *t*. *k* is the multiplication rate of the antigen. The antigen cannot increase absoluteness, because the speed of increase would be affected by the change of density. So we can regard $k(1-\frac{Ag}{Ag_{\max}})$ as the multiplication rate. Along with the antigen increases, the multiplication rate is reduced, and when the density of antigen reaches $Ag_{\max}$, the antigen would not increase anymore. k' is the death rate of cells.

Let $Ag(0)$=0.3 and $\lambda = k - k'$ =3.7, but $Ag(0)$ is 0.3 and 0.301 separately. See fig. 1., from which we can see that if the initial value $Ag(0)$ changed a little, the function would present an obvious different chaos phenomenon.



(a) $Ag(0)=0.3$



(b) $Ag(0)=0.301$

Fig. 1. Chaos phenomena of immune system

3. CIEA：CHAOS IMMUNE EVOLUTIONARY ALGORITHM

Chaos Immune Evolutionary Algorithm (CIEA) has space searching virtues of both Immune Evolutionary Algorithm and Chaos Optimize Algorithm. CIEA imports new "mutation" by chaos operator, and implement "wins and lost" by immune operator. The algorithm has good performance in the aspects of both convergence speed and the global convergence. Fig. 2. shows the workflow of Chaos Immune Evolutionary Algorithm.

The main betterments of the Chaos Immune Evolutionary Algorithm including:

(1) Initial antibodies were generated by chaos sequence that has character of over-spread, randomicity and initial value sensitivity. The data may distribute in solution space equably and not muster together. So the algorithm can overcome data redundancy of random sequence.

(2) Individuals in immune memory were used to vaccinate the other antibodies, which take advantage of the individuals to speed evolution.

(3) Chaos mapping was used to implement mutation, generate new antibody, enlarge search space, and jump out of premature convergence.

(4) Immune selection cannot only ensure high-fitness individuals to evolve, but also restrain high-density individuals.

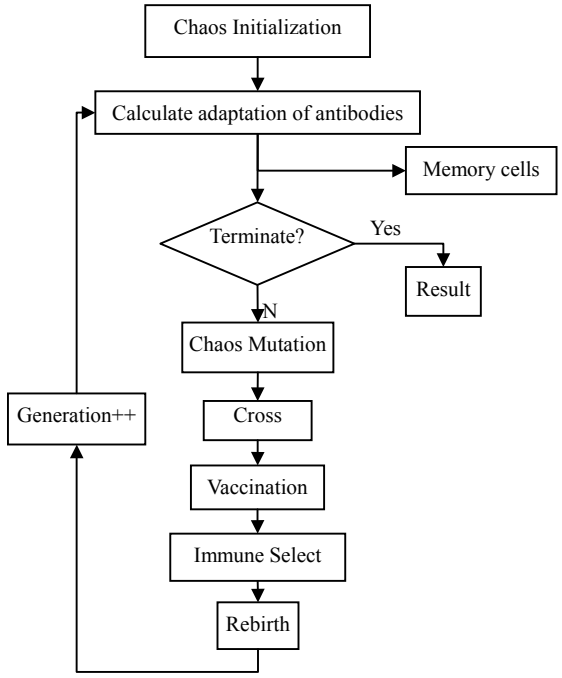


Fig. 2. Flow of Chaos Immune Evolutionary Algorithm

3.1 Chaos Initializations

Because Logistic chaos equation is simple and has a little cost on calculation, CIEA used it to generate chaos sequence:

$$x_{n+1} = \mu x_n (1 - x_n) \tag{2}$$

Where n=1, 2…is the iterative time, μ is a parameter which can control chaos behavior of system, and $0 \le x_n \le 1$, i.e., chaos space is [0,1].

The chaos sequence has the character of over-spread. Chaos sequence was mapped from [0, 1] chaos space to system shape-space and the antibodies were gotten.

It is possible to overcome the data redundancy that using the chaos sequence instead of the random sequence to create initial antibodies, and the antibodies would distribute in system space equably.

Let size of the antibody set is 50, and use random sequence and chaos sequence separately to generate initial candidate antibody set (See Fig. 3.). Symbol "*" denotes the antibodies generated though random sequence, and symbol "o" denotes the antibodies generated though chaos sequence. From Fig. 4., we can see that the chaos sequence has good over-spread character, and can cover the system space uniformly.

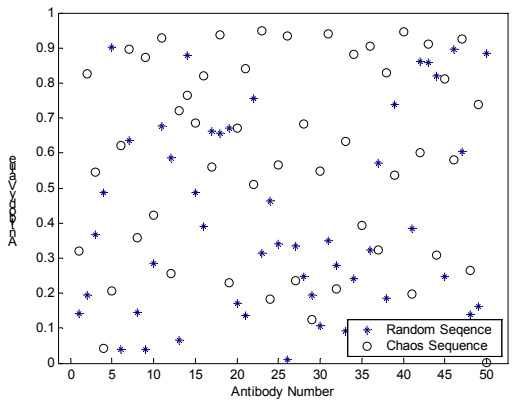


Fig. 3. Chaos Sequence and Random Sequence

3.2 Calculate density and adaptation of antibody

Suppose there are *N* antibodies in antibody set, all antibodies are in *L*-dimension space, and each antibody is an attribute string, which is formed with restricted letters selected form *k*-length alphabet [6]. Equation (3) can calculate information entropy of position j in antibody set.

$$H_j(N) = -\sum_{i=1}^{k} p_{i,j} \log p_{i,j} \tag{3}$$

where $p_{i,}$ is the probability of the ith allele in gene position j. The average information entropy of an antibody set is:

$$H(N) = \frac{1}{L}\sum_{j=1}^{L} H_j(N) \tag{4}$$

The affinity of antibody *v* and *w* can be calculated based on (5), where *H*(2) is the diversity of two antibodies calculated by (4).

$$ay_{v,w} = \frac{1}{1+H(2)} \tag{5}$$

The density of each antibody in antibody set can be presented in (6, where numerator is the number of the antibodies whose affinity with the given antibody is higher than a prearranged threshold value(δ), and denominator is the total number of antibodies.

$$c_v = \frac{1}{N}\sum_{w=1}^{N} ac_{v,w} \tag{6}$$

$$ac_{v,w} = \begin{cases} 1 & ay_{v,w} \geq \delta \\ 0 & otherwise \end{cases} \tag{7}$$

The fitness of an antibody is the affinity of the antibody and all antigens, and can be considered as the evaluation of the problem to be done with.

$$ax_v = fitness(v) \tag{8}$$

3.3 Immune Memory and Vaccination

Antibodies were sorted according to their fitness, and *p* of them with higher fitness would be taken into memory. The size of memory is related to the algorithm performance. If *p* is too small, the algorithm is easy to fall into premature convergence. If *p* is too big, the algorithm speed will not be increased even more. In our algorithm, the size of memory can be dynamically adjusted along with generation number. If *p*=1, the memory cell is the optimal solution of the problem to be done with.

Considering that the space distance of high-fitness individual with optimal solution may be less than that of others with optimal solution and individuals which has less-distance with optimal solution may also has higher fitness, we use memory cells to vaccinate in this text.

$$v' = v_m + \sigma N(0,1) \tag{9}$$

where, *v'* is the vaccine, v_m is the memory cell, *N*(0,1) is a random number obeyed standard normal school, and σ is a adjustable parameter.

Memory cells were used to increase convergence speed in the phase of vaccination, which took advantage of individual to improve the population. Vaccination can make the similar to optimal solution individuals be propagated largely. So the search efficiency and evolution speed would be increased.

3.4 Chaos Mutation

In order to recognize unknown antigen, mutation operators must be used to generate new antibodies. This text ameliorated mutation operator based on chaos mechanism, and use chaos mapping to realize chaos mutation.

Firstly, antibody *v* was mapped into [0,1] space, and taken into (2) as chaos initial value x_0. Secondly, x_0 was iterative looped *r* times, and we can get chaos vector x_r. Thirdly, x_r was mutated based on (10), and the mutated chaos vector x'_r was got. Finally, mapping x'_r into system space, we can get mutated antibody v'.

$$x'_r = (1 - c_v\eta)\hat{x} + \eta x_r \tag{10}$$

where, $\hat{x}$ is current optimal solution, η is a parameter in [0,1] and can be changed automatically along with generation number.

From (10) we can see: because chaos has the character of initial value sensitivity, the mini-difference of x_0 can induce maxi-difference of x_r. This character can enlarge mutation range and optimize mutation. At the same time, the higher the density of antibody is, the wider the search range needed. That is to say, mutation would be the main function in anaphase of evolution, in which the structure of the population tends to be consistent. The higher the range of mutation is, the more easily the algorithm breaks away from premature convergence

3.5 Immune Select and Rebirth

The expectation of antibodies was calculated based on (11) and (12), by which antibodies were selected into next generation.

$$e_v = \frac{ax_v \prod_{j=1}^{N}(1 - as_{v,j})}{c_v \sum_{i=1}^{N} ax_i} \tag{11}$$

$$as_{v,j} = \begin{cases} ay_{v,j}, & ay_{v,j} \geq \gamma \\ 0, & otherwise \end{cases} \tag{12}$$

In order to increase the calculation speed, Equation (13) was used to count the expectation of antibodies in this text.

$$e_v = \frac{ax_v}{c_v} \tag{13}$$

Equation (13) can control the density and diversity of antibodies. Antibodies with high fitness would be selected to rebirth where antibodies with high density would be restrained.

We deal with bacterins as follows: if a bacterin's fitness was less than that of its parent, which shows degenerate phenomena occurred, its parent would be selected but not the bacterin.

Finally, new chaos sequence was generated based on (2) with mini-changed initial value to complement individual number. Because chaos has character of initial value sensitivity, the data redundancy can be avoided.

3.6 Algorithm Detail

```
Function ciea_func()
{
  the initial antibody set was gotten based on (2);
  while(1)
  {
    for every antibody
    {
```

```
        calculate the fitness according to (8);
        record the fitness ;
        calculate the density according to (3)~(7);
    }
    sort the antibodies by their fitness;
    select p antibodies whose fitness were higher than
    others;
    put them into the memory cell set;
    if the stop condition was satisfied
    {
        the algorithm ended;
    }
    lower-fitness individuals were mutated based on (10);
    for every cells in memory set
    {
        the new antibodies were generated based on (9);
    }
     calculate the expectation of every antibody based on
     (13);
    select the individuals whose expectation were higher
    than others, and put them into the next generation;
    new chaos sequence was generated based on (2) with
    mini-changed initial value to complement individual
    number;
  }
}
```

4. EXAMPLE AND ANALYSES

4.1 Tested Function

Several functions were used to check the algorithm capability in this text:

a. Square-sum function

$$f_1(x)=\sum_{i=1}^{3}x_i^2,\quad (-5.12\le x_i\le 5.12); \tag{14}$$

f_1 is a unimodal function presented by Dejong, which reaches minimum at point (0,0,0).

b. Rosenbrock function

$$f_2(x)=100(x_1^2-x_2^2)^2+(1-x_1)^2,\quad (-2.048\le x_i\le 2.048); \tag{15}$$

Although f_2 is a unimodal function, each variable has strong coupling and it is hard to reach minimum. It reach minimum at point (1,1).

c. Rastrigin function

$$f_3(x)=nA+\sum_{i=1}^{n}(x_i^2-A\cos(2\pi x_i)),\quad (-5.12\le x_i\le 5.12); \tag{16}$$

f_3 is a multimodel function, where A is a constant. This function has a great many of local minimums, and reach global minimum at point (0,0,...,0).

d. Griewangk function

$$f_4(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod\cos(\frac{x_i}{\sqrt{i}})+1,\quad (-600\le x_i\le 600); \tag{17}$$

f_4 has thousands of minimums, and the value of the function at point (0,0,...,0) is 0.

4.2 Experiment

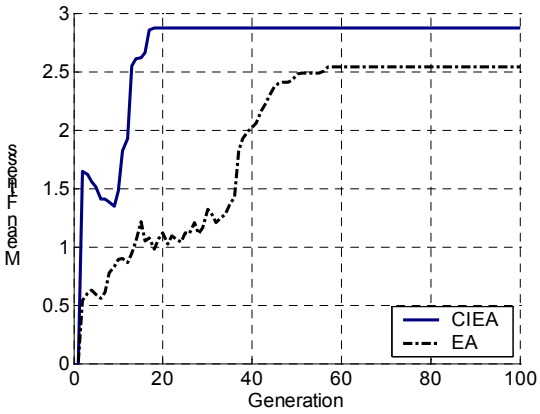
In our experiment, EA and CIEA were used separately to optimize function f_1~ f_4 with same parameter, where cross probability was 0.98, mutation probability was 0.01, and the max fitness change error was 10^{-6}. Let A=10 in function f_3. Table 1. listed the other parameters of our experiment.

Fig. 4. shows the contrast of the two algorithms, where the real line represents the fitness curve of Chaos Immune Evolutionary Algorithm, and the broken line represents the fitness curve of Evolutionary Algorithm.

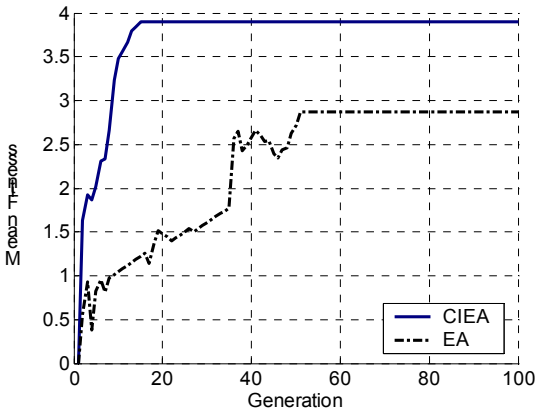
From Fig. 4.(a-d) we can see that the mean fitness of CIEA is higher than that of EA obviously. It is to say that the whole evolve degree of population was improved in CIEA. At the same time it is noticed that CIEA can get optimal solution in less generator, which makes clear that the efficiency of CIEA is higher than that of EA.

Table 1. Experiment parameters

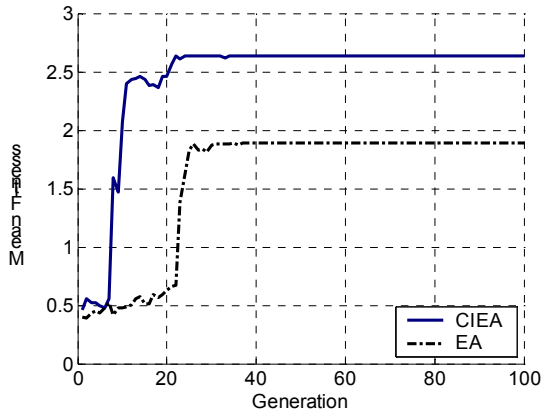
Function	Adaptation Change	Colony Scale	Evolve Generation
f_1	$3.0-\ln(f_1+1.0)$	50	100
f_2	$4.0-\ln(f_2+1.0)$	50	100
f_3	$4.0-\ln(f_3+1.0)$	100	100
f_4	$4.5-\ln(f_4+1.0)$	100	200



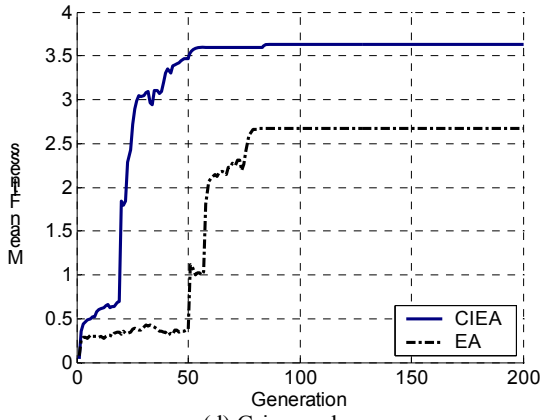
(a) Square-sum



(b) Rosenbrock



(c) Rastrigin



(d) Griewangk

Fig. 4. Contrast of two algorithms mean fitness

5. CONCLUSIONS

Chaos Immune Evolutionary Algorithm used randomicity and space over-spread character of chaos mapping to overcome data redundancy in phase of initialization. In phase of mutation, the initial value sensitivity of chaos was used to enlarge searching area. In addition, local optimal solutions were used to infect filial generation as bacterin, which would keep population diversity, avoid local convergence, and increase operation efficiency.

6. REFERENCES

[1] Q.X.Yun. Evolutionary Algorithm, Metallurgy industry press, Beijing, 2000.

[2] S.Hofmeyr, S.Forrest, “Architecture for an Artificial Immune System,” Evolutionary Computation Journal, 2000，8(4): 443-473.

[3] Leandro N de Castro, Jon Timmis, Artificial Immune System: A Novel Computational Intelligence Approach. Springer-Verlag, 2002.

[4] H.G.Zhang, Z.L.Wang, W.Huang, Control Theory of Chaos System. Publishing House of Northeast University, Shenyang, 2003.9.

[5] A.S.Qi, C.Y.Du, Non-linearity Model of Immune. Shanghai Scientific and Technological Education Publishing House, Shanghai, 1998.

[6] T.Li, Computer Immunology. Publishing House of Electronics Industry, BeiJing, 2004.7.

Parallel Computing and Optimization of CAMS Ocean Model on IA64 Cluster*

Lilun Zhang
National Laboratory of Parallel and Distributed Processing, National University of Defense Technology
ChangSha, Hunan, 410073, China
Email: zll0434@sina.com
Jianglin Hu
National Key Laboratory of Severe Weather, Chinese Academy of Meteorological Science
Beijing, 100081, China
Email: hujl@cma.gov.cn

ABSTRACT

Intel IA64 architecture is very common for high performance computing. On IA64 cluster, parallel computing and optimization of ocean general circulation model from the Research Center of Numerical Forecast, Chinese Academy of Meteorological Science (CAMS) was discussed. Besides model equations and numerical schemes, emphases were put upon model parallelization and performance optimization. Several strategies for optimization were put forward. By using MPI parallel platform, optimization strategies were compared. As a result, we achieve high parallel efficiency with medium-sized parallelism on dual-core Itanium2 IA64 cluster after optimization.

Key words: parallel computing, ocean general circulation model, and performance optimization.

1. INTRODUCTION

CAMS-OGCM is an important component of the climate simulation system of Chinese Bureau of Meteorology, and plays an important role in both understanding the current climate situation and predicting the climate change in the future. Ocean modeling is a computationally intensive job, thus parallel computing is necessary. There are several related efforts to develop parallel ocean general circulation model (POGCM) [1]. Although POGCM have been accepted as a required path to ocean modeling by oceanic physicists, its parallel efficiency is not satisfying in a general way, such as Princeton Ocean Model (POM), OGCM from the State Key Laboratory of Numerical Modeling for Atmospheric Science and Geophysical Fluid Dynamics, Institute of Atmosphere Physics [2]. In this paper, a high-performance parallel ocean model is designed on cluster of dual-core IA64 architecture. In order to performance comparing, we develop three versions of POGCM with different optimization measures based on MPI platform.

The object of our work is the original generation of OGCM from Chinese Academy of Meteorologic Science that is a fully self-made ocean model. This model is to meet the need of developing higher resolution and accuracy large-scale global ocean simulation. It is also an element of the "Research on Development of Climate System" from Chinese Meteorology Agency. We discuss the parallel computing of OGCM on the dual-core IA64 architecture. Emphases are also put upon the optimization of POGCM on this architecture with MPI platform. Accordingly, several strategies for optimization are given and implemented, by which we have obtained approximately liner speedup in POGCM.

The paper is organized as follows. Brief descriptions of the (CAMS-OGCM) ocean model equations are given in Section 2. Section 3 contributes to parallel computing and optimization strategies of POGCM on IA64 architecture. Performance result and discussion are given in section 4. Section 5 contains conclusion and future work.

*National Nature Science Foundation under grant number 40505023 and the Project named "Development of Climate Model System" from Chinese Meteorology Agency support this work.

2. CAMS OCEAN MODEL

The ocean states are characterized by the distribution of currents V , potential temperature T , salinity S , pressure p , and density ρ . Using sphere coordinates with thin shell approximation generates equations that govern the evolution of these fields.

For inner mode (bclinic), the third-order Adams-Bashforth method (AB3) scheme is used for time stepping. Since AB3 is not self-starting, so we have to use lower order schemes for the first two steps, including Euler and second order Adams-Bashforth (AB2) step.

$$\frac{du}{dt} = F(u,t)$$

2^{nd} : AB2 is weakly unstable for convection equation.

$$U^{n+1} = U^n + \Delta t / 2\left[3F^n - F^{n-1}\right]$$

$$U^{n+1} = U^n + \Delta t F^n$$

$$U^{n+1} = U^{n-1} + 2\Delta t F^n$$

$$U^{n+1} = U^n + \frac{\Delta t}{2}\left[3F^n - F^{n-1}\right]$$

3^{nd} :

$$U^{n+1} = U^n + \frac{\Delta t}{12}\left[23F^n - 16F^{n-1} + 5F^{n-2}\right]$$

For outer mode (barotr), simple forward Euler scheme is adopted.

3. PARALLEL COMPUTING and OPTIMIZATION

We utilize MPI platform to parallelize OGCM with SPMD parallel programming mode. For this mode, data distribution is the key problem. Data splitting follows several guidelines: (1)computational load balance; (2)maxium local computation and minal communication; (3) no unnecessary superfluous computation. It is desirable to spitting data by latitude zone to get a satisfied load balance. The reason for one-dimension data spitting is the consideration of communication. Since we use Adams-Bashforth (AB3) and forward Euler schemes for time discrezation. They are all explicit difference schemes; it is necessary for data zones on logically adjacent computing processors to have overlapping boundaries in SPMD parallel computing. In applications, boundary overlapping is implemented through message passing between neighboring processors. We call it neighboring communication.

Steps of ocean model computation:

step1.
Parallel initialization, generate local index boundaries of echo parallel task.
step2.
The root process deal with data IO, and scatter data to corresponding task.
step3.
Time stepping of ocean model
{
3.1Loop of inner mode
{
3.11 Loop of outer mode
{
time stepping of barot integration
boundaries communication
}
3.12 time stepping of bclin. integration
3.13 coupling of barotr and bclinic
3.14 updating of vertical variables
3.15 boundaries communication
}
3.2 Thermodynamics step
3.3 Updating of state variables
}
step4.
Gathering data and writing history file

Measurements of performance optimization:

Memory and Cache Optimization is critical to efficient utilization of the memory centric cluster. Because of the great disparity between the central processing unit (CPU) and the memory subsystem speed, “memory wall” seems to be the biggest obstacle to make full use of the high performance computer system. As a result, application performance is to a great extent dominated by memory access time. To deal with the “memory wall”, cache is of great importance. Several strategies are designed and implemented to improve locality for efficient memory access. In general, there are two means to enhance locality: data locality inside loop and data dependency between loops [3]. To improve locality inside loop, data accessed in a same loop should less than cache capacity, and the accesses of the array are modified to be continuous to the great extent. Strategies designed for this goal are as follows.

1. Some new arrays are adopted in computing, whose sizes decrease linearly with the increase of processor number. On the other hand, the original big arrays are reserved to record the history files.
2. Loop containing too many variables is split into smaller loops, in case it is separable.
3. Some nested loops are reordered for continuous memory access. For the problem of locality between loops, loops with one or more related variables are made closer, which is called aggregation..

Redundancy Computing is a natural choice to attain good performance. Since redundant computations are both waste and inclined to cause errors in statistic computing, we should take care of it. Update of variables should be restricted to the data inside the absolute boundaries, and conditional sentences in parallel loops are replaced by predefined relative boundaries.

Communication Optimization For both distributed memory parallel and shared memory parallel computing, the communication overhead is far more than a CPU clock cycle. Communication optimization is critical. (1) Asynchronous communication is combined with communication adjustment. Asynchronism is to make full use of the communication bandwidth, and communication adjustment is adopted properly to make sure there is no message jam. That is, the whole nodes can be divided into two-even-odd sets, nodes in the same set never do “send” and “receive” operation simultaneously. (2) Aggregation of short messages. Because the communication time of short messages is decided by the system overhead, the time to send message to or receive it from the network, aggregation communication can save time by reduce the frequency of message passing.(3)Recursive method for data gather based on the “two-divide” idea.

4. TEST RESULTS

We implemented POGCM on Intel IA64 linux cluster. FORTRAN compiler is Intel ifort and MPI implementation is MPICH2.097. Model mesh size is 360*130*33. Although performance model of parallel ocean model has been put forward [4, 5] recently, it is still hard to find precise value by the performance model. So it’s not feasible to compare parallel performance by mathematics model. We have developed three version of CAMS ocean model with different parallel and serial optimization measures. And we make it sure that the three versions have absolutely identical computational results by correct modification and using compiler flags “–mp –IPF_fltacc”. The three versions of CAMS ocean model are as follows:

Version1 (V1): the original parallel version of CAMS Ocean model based on MPICH, no optimization.

Version2 (V2): optimization with redundancy Avoiding, communication aggregation.

.

Version3 (V3): with communication aggregation, loop sequence changed.

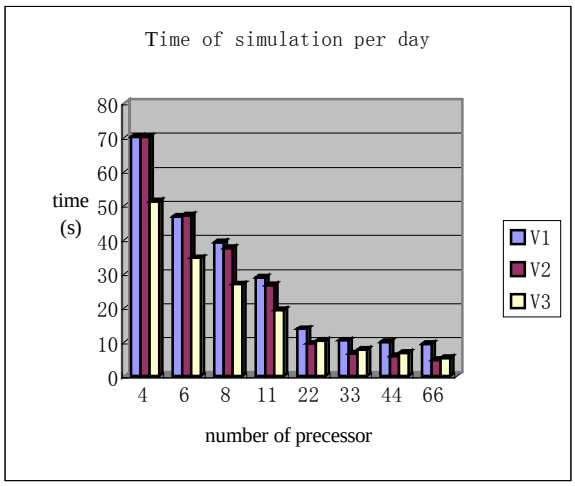


Fig. 1. Time used for simulation per day

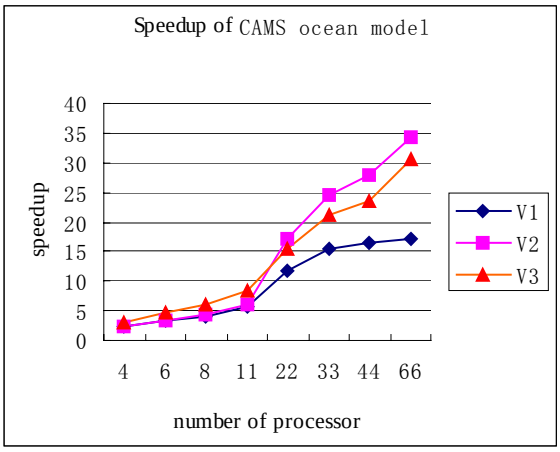


Fig. 2. Speedup of three versions

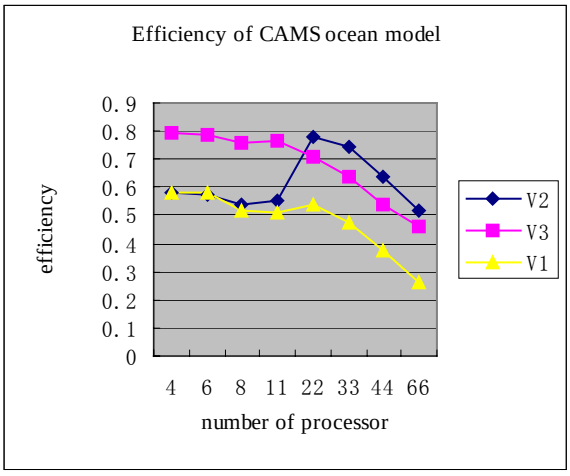


Fig. 3. Parallel efficiency of three versions

Fig.1. shows three versions with different optimization strategies running in a far cry manner and optimization plays an important role. Fig. 2. shows that V3 running an advantage position with small parallel size than V2, but declines remarkably with the increase of parallel size. Fig. 3. shows that V2 has higher parallel efficiency which drops slowly, and V3 has poor efficiency at the beginning, which increases abruptly and then declines as quickly as V2.

5. CONCLUSION

CAMS Ocean general circulation model is parallelized. Several optimization strategies are adopted for performance improvement. As a result, three versions of parallel model are generated with different running manner on IA64 cluster. In future, OpenMP will be used for optimization.

6. REFERENCES

[1] Dale B.Haidvogel, Aike Beckmann, “Numerical Ocean Circulation Modeling”, Beijing, Meteorology Publishing Company, 2005.
[2] H.L. Liu, Y.Q. Yu, et al, Reference Handbook of LICOM1.0 for Ocean model in Climate System, Beijing: Science Publishing House, 2003.
[3] H.Q. Ding, Y. He, Data Organization and I/O in a Parallel Ocean Circulation Model, Proceeding of Supercomputing, Portland, Oregon, November 1999.
[4] Darren J. Kerbyson, “A Performance Model of the Parallel Ocean Program”, International Journal of High Performance Computing Applications, 2005, 19(3): 261-276.
[5] L.L. Zhang, H. Ye, et al, “Optimal Overlapping Boundaries Model for Distributed Memory Parallel Computing of Difference Discrete”, Journal of National University of Defense Technology, 2000, 22(5): 64-68.

Solving Traveling Salesman Problem by Ant Colony Optimization -Particle Swarm Optimization Algorithm

Shang Gao1,2
1School of electronics and information, Jiangsu University of Science and Technology, Zhenjiang 212003,China;
2Provincial Key Laboratory of Computer Information Processing Technology, Soochow University, Suzhou 215006,China
Email: gao_shang@hotmail.com
Ling-fang Sun
School of Economic and management, Jiangsu University of Science and Technology, Zhenjiang 212003,China
Email: slf0308@163.com
Xin-zi JIANG
School of electronics and information, Jiangsu University of Science and Technology, Zhenjiang 212003,China
Ke-zong TANG
School of electronics and information, Jiangsu University of Science and Technology, Zhenjiang 212003,China

ABSTRACT

Using the properties of ant colony algorithm and particle swarm optimization, a hybrid algorithm is proposed to solve the traveling salesman problems. First, it adopts statistics method to get several initial better solutions and in accordance with them, gives information pheromone to distribute. Second, it makes use of the ant colony algorithm to get several solutions through information pheromone accumulation and renewal. Finally, using across and mutation operation of particle swarm optimization, the effective solutions are obtained. Compare with the simulated annealing algorithm, the standard genetic algorithm and the standard ant colony algorithm, all the 16 hybrid algorithms are proved effective. Especially the hybrid algorithm with across strategy B and mutation strategy B is a simple and effective better algorithm than others.

Keywords: Ant Colony Algorithm, Particle Swarm Optimization, Traveling Salesman Problem, Simulated Annealing Algorithm, Genetic Algorithm.

1. INTRODUCTION

Inspired by the behavior of real ants, Marco Dorigo first introduced the colony optimization approach in his Ph.D. thesis in 1992 and expanded it in his further work. The characteristics of artificial ant colony include a method to construct solutions that balances pheromone trails and a problem-specific heuristic, a method to both reinforce and evaporate pheromone, and local search to improve the constructed solutions. The ACO[1] methods have been successfully applied to diverse combinatorial optimization problems including traveling salesman, quadratic assignment, vehicle routing[2], telecommunication networks[3], graph coloring, constraint satisfaction, Hamitonian graphs and scheduling. Particle swarm optimization is an evolutionary computation technique (a search method based on a natural system) developed by Kennedy and Eberhar in 1995. Particle Swarm Optimization (PSO)[4,5,6] is an optimization technique inspired by social behavior observable in nature, such as flocks of birds and schools of fish. It is essentially a nonlinear programming technique suitable for optimizing functions with continuous domains (though some work has been done in discrete domains), and has a number of desirable properties, including simplicity of implementation, scalability in dimension, and good empirical performance. The PSO algorithm was originally proposed for continuous problems, and the hybrid algorithm combining ACO algorithm have been made recently to extend it to discrete optimization problems such as traveling salesman problem (TSP).

2. THE BASIC ACO ALGORITHM

In this section we introduce the basic ACO algorithm. We decided to use the well-known traveling salesman problem as benchmark, in order to make the comparison with other heuristic approaches easier. Given a set of n towns, the TSP can be stated as the problem of finding a minimal length closed tour that visits each town once. We call d_{ij} the length of the path between towns i and j. In the case of Euclidean TSP, d_{ij} is the Euclidean distance between i and j (i.e., $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. An instance of the TSP is given by a graph (N, E), where N is the set of towns and E is the set of edges between towns (a fully connected graph in the Euclidean TSP).

Let $b_i(t)$ ($i = 1,2,\cdots,n$) be the number of ants in town i at time t and let $m = \sum_{i=1}^{n} b_i(t)$ be the total number of ants. Each ant is a simple agent with the following characteristics:

- it chooses the town to go to with a probability that is a function of the town distance and of the amount of trail present on the connecting edge.
- to force the ant to make legal tours, transitions to already visited towns are disallowed until a tour is completed (this is controlled by a *tabu* list).
- when it completes a tour, it lays a substance called trail on each edge (i, j) visited.

Let $\tau_{ij}(t)$ be the intensity of trail on edge (i, j) at time t. Each ant at time t chooses the next town, where it will be at time $t+1$. Therefore, if we call an iteration of the ACO algorithm the m moves carried out by the m ants in the interval $(t, t+1)$, then every n iterations of the algorithm (which we call a cycle) each ant has completed a

tour. At this point the trail intensity is updated according to the following formula

$$\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \quad (1)$$

where ρ . is a coefficient such that (1- ρ) represents the evaporation of trail between time t and $t+n$,

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (2)$$

where $\Delta\tau_{ij}^{k}$ is the quantity per unit of length of trail substance (pheromone in real ants) laid on edge (i, j) by the k-th ant between time t and $t+n$. It is given by

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{if k - th ant uses edge (i, j) in its tour (between time t and t + n)} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where Q is a constant and L_k is the tour length of the k-th ant.The coefficient ρ must be set to a value ρ <1 to avoid unlimited accumulation of trail. In our experiments, we set the intensity of trail at time 0, $\tau_{ij}(0)$, to a small positive constant *c*.

In order to satisfy the constraint that an ant visits all the n different towns, we associate with each ant a data structure called the *tabu list*, that saves the towns already visited up to time t and forbids the ant to visit them again before n iterations (a tour) have been completed. When a tour is completed, the *tabu* list is used to compute the ant's current solution (i.e., the distance of the path followed by the ant). The *tabu* list is then emptied and the ant is free again to choose. We define $tabu_k$ the dynamically growing vector, which contains the *tabu* list of the kth ant, $tabu_k$ the set obtained from the elements of $tabu_k$, and $tabu_k(s)$ the s-th element of the list (i.e., the s-th town visited by the k-th ant in the current tour).

We call *visibility* η_{ij} the quantity $1/d_{ij}$. This quantity is not modified during the run of the ACO algorithm, as opposed to the trail, which instead changes according to the previous formula (1).

We define the transition probability from town i to town j for the k-th ant as

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}}{\sum_{s\in allowed_k}\tau_{is}^{\alpha}(t)\cdot\eta_{is}^{\beta}} & if \ \ j \in allowed_k \\ 0 & otherwise \end{cases} \quad (4)$$

where $allowed_k$ = { N - $tabu_k$ } and where α and β are parameters that control the relative importance of trail versus visibility. Therefore the transition probability is a trade-off between visibility (which says that close towns should be chosen with high probability, thus implementing a greedy constructive heuristic) and trail intensity at time t (that says that if on edge (i, j) there has been a lot of traffic then it is highly desirable, thus implementing the autocatalytic process).

3. THE BASIC PSO ALGORITHM

In the PSO algorithm, the birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility.

The velocity and direction of each particle moving along each dimension of the problem space will be altered with each generation of movement. In combination, the particle's personal experience, $\boldsymbol{P_{id}}$ and its neighbors' experience, $\boldsymbol{P_{gd}}$ influence the movement of each particle through a problem space. The random values $\boldsymbol{rand_1}$ and $\boldsymbol{rand_2}$ are used for the sake of completeness, that is, to make sure that particles explore a wide search space before converging around the optimal solution. The values of $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ control the weight balance of $\boldsymbol{P_{id}}$ and $\boldsymbol{P_{gd}}$ in deciding the particle's next movement velocity. At every generation, the particle's new location is computed by adding the particle's current velocity, $\boldsymbol{v_{id}}$, to its location, $\boldsymbol{x_{id}}$. Mathematically, given a multi-dimensional problem space, the ***ith*** particle changes its velocity and location according to the following equations:

$$v_{id} = w\times v_{id} + c_1\times rand_1\times(p_{id} - x_{id}) + c_2\times rand_2\times(p_{gd} - x_{id}) \quad (5)$$

$$x_{id} = x_{id} + v_{id} \quad (6)$$

where $\boldsymbol{w}$ denotes the inertia weight factor; $\boldsymbol{p_{id}}$ is the location of the particle that experiences the best fitness value; $\boldsymbol{P_{gd}}$ is the location of the particles that experience a global best fitness value; $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are constants and are known as acceleration coefficients; $\boldsymbol{d}$ denotes the dimension of the problem space; $\boldsymbol{rand_1}$, $\boldsymbol{rand_2}$ are random values in the range of (0, 1).

4. HYBRID ALGORITHM

4.1 Modified ACO Algorithm

In traditional ACO Algorithm, the initialization of the pheromone matrix is equal. Ants need iterate many numbers to find the best tour. We can generate a large amount of tours (e. g. 100 tours), and then we choose some better tours (e. g. 30 tours). At last ants lay trail only on these better tours. These trails affect following ants.

When ant completes a tour, it always lays called trail on each edge visited. If the tour is worse, ant also lay the trail on each edge. These trails disturb the following ants, so the ACO algorithm's convergent speed is very slow. We can calculate the length of tour firstly. and then we compare with the given value. If the length of tour is less than the given value, we update trail values. Otherwise don't update the trail values.

The advantage of PSO algorithm method is that it use self information, individual best information and global best information. We can learn from genetic algorithm, and we rearrange formula (5). The $w\times v_{id}$ can be regarded as mutation operator, and $c_1\times rand_1\times(p_{id} - x_{id}) +$

$c_2 \times rand_2 \times (p_{gd} - x_{id})$ can be regarded as crossover operator. But the crossover operator happened between the individual with local optimum and global optimum.

4.2 Mutation operators

There are following methods to generate a new tour C_1 from the tour C_0.

Mutation operator A Choose two cities j_1 and j_2 from the tour C_0 by randomly, and then swap j_1 with j_2 in the tour C_0, so the new tour is C_1. For example, suppose C_0=2 3 4 1 5 7 9 8 6, $j_1 = 3$ and $j_2 = 9$, so C_1=2 9 4 1 5 7 3 8 6.

Mutation operator B Choose a city j_1 from the tour C_0 by randomly, and then swap j_1 with the next visited city. For example, suppose C_0=2 3 4 1 5 7 9 8 6, $j_1 = 3$, so C_1=2 4 3 1 5 7 9 8 6.

Mutation operator C A modified solution C_1 is generated from C_0 by randomly choose two cities j_1 and j_2 and reversing the sequence in which the cities in between cities j_1 and j_2 are traversed, i.e. the 2-change generation mechanism. For example, suppose C_0=2 3 4 1 5 7 9 8 6, $j_1 = 3$ and $j_2 = 9$, so C_1=2 9 7 5 1 4 3 8 6.

Mutation operator D Choose two cities j_1 and j_2 from the tour C_0 by randomly, and then insert city j_1 into the latter of j_2 city. For example, suppose C_0=2 3 4 1 5 7 9 8 6, $j_1 = 3$ and $j_2 = 9$, so C_1=2 4 1 5 7 9 3 8 6.

4.3 Crossover Operators

There are many different types of crossover operators, but we discuss several usual crossover operators as following. Let's suppose we have two parent tours given by

old1=1 2 3 4 5 6 7 8 9

old2=9 8 7 6 5 4 3 2 1.

Crossover operators A We will swap a substring from old2 that will be called the donor, and this substring is selected randomly. Then we insert the substring into the beginning (or the end) of old1. After the insertion of the substring we must delete the cities of the receptor that were included in the new substring added. For example the substring 6 5 4 3 of donor old2 is the one that will be inserted into the beginning of old1,and we delete the cities of the receptor that were included in the new substring added. After that, we get the configuration

new1=6 5 4 3 1 2 7 8 9.

Crossover operators B We will swap a substring from old2 randomly. Then we insert the substring into the matching section of old1. After the insertion of the substring we must delete the cities of the receptor that were included in the new substring added. For example the substring 6 5 4 3 is from donor old2, so after rearrangement we have

new1=1 2 6 5 4 3 7 8 9.

Crossover operators C We will swap a substring from old2 randomly. Then we insert the substring into the random section of old1. After the insertion of the substring we must delete the cities of the receptor that were included in the new substring added. For example the substring 6 5 4 3 is from donor old2, and if city 7 is the random city of old1, so after rearrangement we have new1=1 2 7 6 5 4 3 8 9.

Crossover operators D The substring 6 5 4 3 is from donor old2 randomly, then we insert the substring into the position of city 6 of old1, and we delete the cities of the receptor that were included in the new substring added，so new1=1 2 7 6 5 4 3 8 9.

4.4 Hybrid Algorithm

The hybrid algorithm solving TSP can be expressed as follows:

Step1. $nc \leftarrow 0$ （nc is iteration number）. Generate 100 tours, and choose the better 30 tours from these 100 tours, and pheromone laid on edge of these 30 better tours.

Step 2. *Pbests* is set to initial tours and the initial best evaluated value among *pbests* is set to *gbest*.

Step 3. Choose the next city j according to formula (4).

Step 4. The j th tour $C_0(j)$ and *pbest(j)* are required to crossover to generate an offspring $C_1^{'}(j)$. $C_1^{'}(j)$ and the *gbest* are required to crossover to generate an offspring $C''_1(j)$, and then $C''_1(j)$ is mutated to $C_1(j)$ according to the mutation probability. Calculate the new evaluation values.

Step 5. If the evaluation value of each ant is better than the previous *pbest*, the value is set to *pbest*. If the best *pbest* is better than *gbest*, the value is set to *gbest*.

Step 6. Compute L_k $(k = 1,2,\cdots,m)$ (L_k is the length of tour done by ant k). Save the current best tour.

Step 7. If the L_k is less the given value, update trail values according to formula (1), formula (2) and formula (3).

Step 8. $nc \leftarrow nc + 1$.

Step 9. If the iteration number nc reaches the maximum iteration number, then go to Step 9. Otherwise, go to Step 2.

Step 10. Print the current best tour.

5. EXPERIMENTAL RESULTS

This section compares the results of simulated annealing algorithm, genetic algorithm, ACO algorithm and hybrid algorithms on traveling salesman problem of 30 cities. The parameters of simulated annealing algorithm are set as follows: the initial temperature $T = 100000$, the final temperature $T_0 = 1$, and annealing velocity $\alpha = 0.99$. The parameters of the genetic algorithm optimization toolbox (GAOT) used to solving TSP are set as follows: the population $N = 30$, the cross probability $P_c = 0.2$, and the mutation probability $P_m = 0.5$. The parameters of the

hybrid algorithms are set as follows: $\alpha = 1.5$, $m = 30$, $\beta = 2$, and $\rho = 0.9$. 20 rounds of computer simulation are conducted for each algorithm, and the results are shown in Table 1. The optimal tour of 30 cities by hybrid algorithm is shown in Fig. 1. All the 16 hybrid algorithms are proved effective. Especially the hybrid algorithm with across strategy B and mutation strategy B is a simple and effective better algorithm than others. The crossover operators of particle swarm optimization are different from those of genetic algorithm. In hybrid optimization, the crossover operator happened between the individual with local optimum and global optimum, so the capability of offspring got to be improved.

Table 1. Testing result of algorithms

Algorithms	Average solutions	Best solutions	Worst solutions
Aimulated annealing algorithm	438.5223	424.6918	479.8312
Genetic algorithm	483.4572	467.6844	502.5742
Basic ACO algorithm	550.0346	491.9581	599.9331
Crossover operators A +Mutation operator A	439.4948	425.6490	456.7721
Crossover operators A +Mutation operator B	441.9257	428.7296	455.2382
Crossover operators A +Mutation operator C	437.0028	426.6002	446.2394
Crossover operators A +Mutation operator D	438.7750	425.4752	455.2929
Crossover operators B +Mutation operator A	438.9350	424.6354	457.9062
Crossover operators B +Mutation operator B	431.4987	423.7406	447.6865
Crossover operators B +Mutation operator C	435.4220	424.9003	447.3223
Crossover operators B +Mutation operator D	439.4777	426.1972	465.9935
Crossover operators C +Mutation operator A	444.1723	429.3803	459.4925
Crossover operators C +Mutation operator B	438.5871	426.3076	455.5854
Crossover operators C +Mutation operator C	440.4201	427.6016	454.8674
Crossover operators C +Mutation operator D	439.5524	424.4643	461.7948
Crossover operators D +Mutation operator A	439.0477	424.6727	451.8001
Crossover operators D +Mutation operator B	436.0081	423.7406	460.6230
Crossover operators D +Mutation operator C	438.8091	425.8201	455.4830
Crossover operators D +Mutation operator D	436.4577	423.9490	457.3155

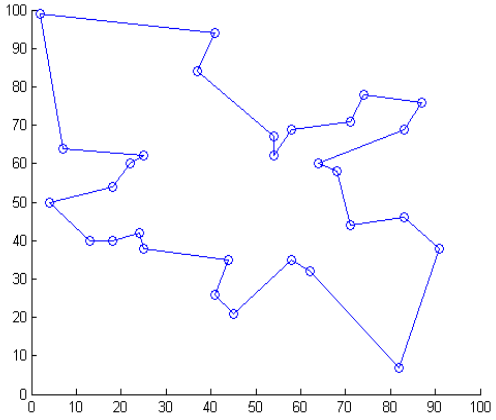


Fig. 1. The optimal tour of 30 cities by hybrid algorithm

6. CONCLUSIONS

The classical PSO is a powerful method to find the minimum of a numerical function, on a continuous definition domain. This paper presents a hybrid ACO algorithm with PSO for solving the well known TSP. Experiments for benchmark problems show the hybrid algorithm better than other algorithms.

The following problems need to be considered. 1. The parameters and their affect on the performance of the optimization should be studied in more detail. 2. How to explore hybrid algorithm application to continuous space problem should be investigated. 3. The hybrid algorithm's convergent speed, or the efficiency, should be worth further investigating. 4. How to evaluate the quality of hybrid algorithm and other algorithms is still a problem.

REFERENCES

[1] A. Colorni, M. Dorigo, et al "An investigation of some properties of an ant algorithm". Proc. Of the Parallel Problem Solving from Nature Conference (PPSN'92)[C]. Brussels, Belgium: Elsevier Publishing, 1992, 509-520.

[2] B.Bullnheimer, F. R. Hartl, et al. "Applying the ant System to the vehicle routing problem". 2nd Metaheuristics International Conference (MIC-97)[C]. Sophia-Antipolis, France, 1997, 21-24.

[3] G. D. Caro, M. Dorigo. "Mobile agents for adaptive routing". Proceedings of the 31th Hawaii International Conference on system Sciences. Big Island of Hawaii, 1998, 74-83.

[4] R. C. Eberhart. , J. Kennedy, "A new optimizer using particles swarm theory". Proc. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, 39-43.

[5] Y. H. Shi, R. C. Eberhart, "A modified particle swarm optimizer". IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998, 69-73.

[6] E. Ozcan, C. Mohan. "Particle swarm optimization: Surfing the waves". Proc of the Congress on Evolutionary Computation. Washington DC, 1999, 1939-1944.

A Hybrid GA/CMAC Algorithm for Improving the Accuracy in Modeling of Function Approximation

Xin Tan
School of Communication, Chongqing University of Posts and Telecommunications
Chongqing 400065, China
Email: weisite2008@yahoo.com.cn

And

Huaqian Yang
Department of Computer and Modern Education Technology, Chongqing Education College
Chongqing 400067, China
Email: mailtostorm@sohu.com

ABSTRACT

The structure of the Cerebellar Model Articulation Controller (CMAC) is based on the operation of the human cerebellum with good characteristics of generalization and fast learning property. Owing to these attractive properties, the CMAC can be used to various application areas, such as approximate nonlinear functions, pattern recognition, and so on. In this paper, Genetic Algorithms is employed on the training of CMAC, which can optimize parameters of the CMAC. To demonstrate the effectiveness of the proposed method, simulation experiment results were illustrated. The results show that, in function approximation, the proposed strategy has better accuracy than the conventional CMAC.

Keywords: CMAC, Neural Networks, Genetic Algorithms, Approximate, Optimize.

1. INTRODUCTION

The cerebellar model arithmetic controller (CMAC) is an adaptive system that was first proposed by Albus in 1975 [1]. Albus proposed a control method that is based on the principles of the cerebellum's motor behavior; he called the control system CMAC. It is a typical kind of local neural network and can be used in control learning, implementing a mapping or function approximation.

The CMAC is employed to approximate and compensate the uncertainties induced by inaccurate modeling of the system to be controlled. The CMAC combines the input commands and output feedback variables into an input vector that is used to address the memories where the related values are stored.

To use CMAC, we have to decide various parameters depending on problems to solve. However the guideline for parameters setting of CMAC has not been established yet. Genetic Algorithms (GA) is promising methods for this search for the optimal boundaries of subspaces or optimal positions and size of granule cells of CMAC. In addition, since the conventional CMAC uses a constant basis function as association memory selection vector to model the hypercube structure, its output is always constant within each quantized state and the derivative information of input and output variables cannot be preserved [2, 3]. The non-differentiable property leads to some limitations when using the conventional CMAC in real-world applications, such as action-dependent critical learning. Therefore, we proposed an effective hybrid strategy: Genetic Algorithms combining with cerebellar model Arithmetic controller (GA/CMAC) to improve the accuracy in modeling of function approximation.

A brief introduction of some basic concepts of the CMAC and Genetic Algorithms was made in section 2 and section 3. Then GA/CMAC was described in Section 4. Experiment results are shown in section5. Eventually, a brief conclusion is depicted in Section 6.

2. BASIC CONCEPTS OF THE CMAC NETWORK

The CAMC is conceptually simpler, has faster learning convergence, and more practical than the well-known feed forward back propagation artificial neural network architectures. The Structure of CMAC is shown in Fig.1.

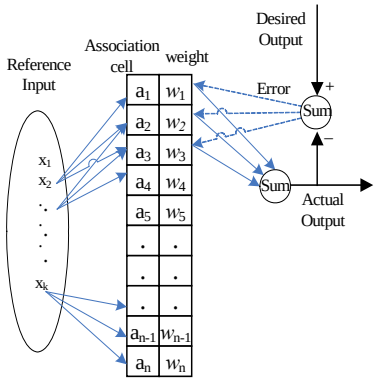


Fig. 1. The Structure of CMAC

CMAC is a learning and storage structure that initiates the human cerebellum. Generally speaking, it can be viewed as a lookup table. In CMAC, each state variable is quantized and input space is divided into discrete states. Information for a state is distributive stored in memory using state variables as an address key and then into physical memory addresses. The addresses are used to fetch and compute the output. Output of the CAMC is computed through a sequence of mapping. It is computed from a linear combination of weights that are stored in memory. In CMAC-based learning systems, selection of the CMAC structural parameters is important. Such parameters determine how the CMAC input space is quantized and how

the memory is used.

CMAC has the local generalization abilities that are dependent on the overlap of the association vector by mapping algorithm. The principle is used to give the address of the weights that contribute to output for similar inputs, which is "the function with similar outputs for similar inputs". So the mapping algorithm is critical to CMAC. The dependence of the adjustable parameters or the weights with respect to corresponding outputs is linear, and many learning algorithm are applicable. The selection of the learning algorithms must be considered.

The mapping of CMAC can be divided into two parts: $X \rightarrow A$ and $A \rightarrow Y$.

Where X: continuous k-dimensional input space; A: n-dimensional association space; Y: one-dimensional output. The first mapping, function $S(x)$ maps each point x in the input space onto the association vector that has n nonzero elements. For a standard CMAC model, the association vector contains only binary elements, either zero or one. The second mapping function $P(a)$ computes a scalar output y by projecting the association vector onto a vector w of adjustable weights so that the scalar output can be easily obtained by evaluating the inner product of the two vector a and w.

$$y = P(a) = a^T w \tag{1}$$

The weights will be updated by the learning rule as follows:

$$w(i+1) = w(i) + \Delta w = w(i) + \frac{\alpha}{m} a^T (y_d - a^T w(i)) \tag{2}$$

Where i is the ith iteration, α is learning rate, m is the number of activated association cells, and y_d is the desired output.

3. THE PROCESS OF GENETIC ALGORITHMS

Genetic algorithms have been gaining popularity in a variety of applications, which require global optimization of a solution. A good general introduction to genetic algorithms is given in [4]. The Genetic Algorithms are allowed to select subsets of various sizes to determine the optimum combination and number of inputs to the network. The emphasis on using the Genetic Algorithms for feature selection is to reduce the computational load on the training system while still allowing near optimal results to be found quickly.

First, Binary-string encoding [5] which is the most classic approach due to its simplicity and traceability is often used. Second, to maintain uniformity over various problem domains, a fitness function is needed to map the objective value to a fitness value. Third, The usual selection operator is the roulette wheel selection that is individual with high fitness has more chances to be selected. Firth, it includes tow types of genetic operators. Crossover is a recombination operator by crossover probability Pc that combines subparts of two parent chromosomes to produce offspring that contain some parts of both parents' genetic material. There are tow main crossover operators such as single-point crossover operator, double-points. The mutation operator is always used to keep the diversity of population which independently employs the logical not operation by mutation probability Pm. After generating the offspring, the generational replacement may make the best member of the population fail to reproduce offspring in the next generation. So the method may increase the speed of domination of a population by a super chromosome, but on balance it appears to improve the performance.

4. GA/CMAC WITH GAUSSIAN BASIS FUNCTION

GA can be used to adjust the partition in each layer of CMAC for identifying an appropriate model. Genetic Algorithms is applied to search for appropriate partitions of CMAC. CMAC has a fast learning capability. But the partitions in each layer have been usually fixed. CMAC has needed many layers or granule cells to identify accurate models. The boundaries of subspaces should be modified in the learning phase to decrease the layers and the granule cells.

Then, we try to apply a Gaussian basis function to GA/CMAC. The conventional CMAC uses local constant basis function. If the constant basis functions are replaced by non-constant differentiable basis function, the derivative information will be able to be stored into the structure as well.

The generalized CMAC technique uses general basis function [6, 7]. The basis function $b_i(\cdot)$ is associated with the ith hypercube. All basis functions have bounded values inside their hypercube areas. Note that if $b_i(\cdot)$ is a constant inside the area covered by the ith hypercube and zero outside, the generalized scheme becomes the conventional CMAC. To determine an output value for a given input, a linear combination of the basis function associated with involved hypercube is used [6].

If the basis functions are differentiable, the output derivative can be preserved. On the other hand, compared with general basis functions network (GBFN), with the CMAC local selection scheme, the number of basis function involved for each given input in the new technique is quite little. This can significantly reduce the computational load. It is noted that the means and variances of the Gaussian functions can also be adjusted to increase the approximation capability.

5. EXPEREMENT RESULTS

In this section, we use conventional CMAC, GA/CMAC and GA/CMAC with GBF for identifying the input-output relationships of the data generated from the following two functions: $\hat{Y}_{d,1}$ and $\hat{Y}_{d,2}$.

$$\hat{Y}_{d,1} = y^2 \cos(x) + x^2 \sin(y) \quad x = [-1,1], y = [-1,1] \tag{3}$$

$$\hat{Y}_{d,2} = xy \exp(-(x^2 + y^2)) \quad x = [-1,1], y = [-1,1] \tag{4}$$

The constructed CMAC consisted of four layers with two inputs and one output. Input spaces of x and y are quantized into 10 states and each complete block has 3 states. There are totally 48 weights (4*4*3=48). The learning rate is set for 0.4. The stop criterion is set for 60 learning cycles. The locations of partitions of the CMAC were encoded into the chromosomes by binary bit. Ten chromosomes are generated in a population. We compare the Mean Square Error (MSE) in Fig.2 and Fig.3.

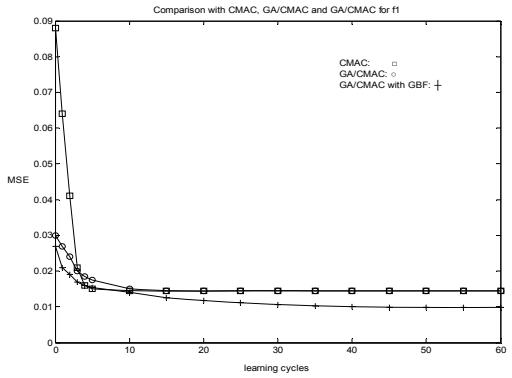


Fig. 2. Comparison with CMAC, GA/CMAC and GA/CMAC with GBF for $\hat{Y}_{d,1}$

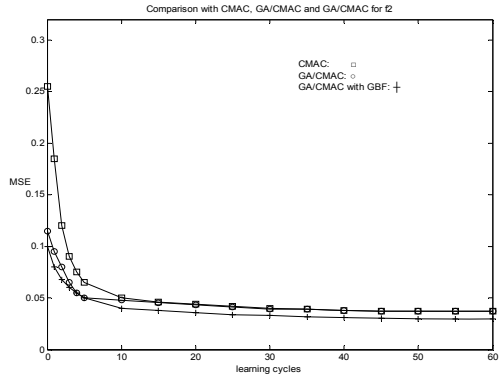


Fig. 3. Comparison with CMAC, GA/CMAC and GA/CMAC with GBF for $\hat{Y}_{d,2}$

6. CONCLUSIONS

In this paper, we have presented hybrid GA/CMAC with GBF for function approximation applications. GA is a global search method, which has the ability to globally approach the optimization solutions. We use GA to optimize parameters of the CMAC. Then, Gaussian basis function is applied to GA/CMAC to improve the accuracy in function approximation. From the simulation results, the performance of GA/CMAC with GBF has been compared with the conventional CMAC. This results show that the proposed method has smaller Mean Square Error than the traditional CMAC.

7. REFERENCES

[1] J. S. Albus, Brains, Behavior and Robotics, McGraw-Hill, 1981.

[2] C. T. Chiang, C. S. Lin, “CMAC with general basis functions”, Neural Networks, 9(7), 1996,1199–1211.

[3] C. S. Lin, C. T. Chiang, “Learning convergence of CMAC technique”, IEEE Transactions on Neural Networks, vol. 8, no.6, 1997,1281-1292.

[4] K. A. Tang, K. F. Man, et al, “Genetic algorithms and their applications”, IEEE Signal Process, 1996,22-37.

[5] H. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor, The University of Michigan Press, 1975.

[6] A. Kolcz, N. M. Allinson, “Basis function models of the CMAC network”, Neural Networks, 12(1), 1999,107-126.

[7] H. M. Lee, C. M. Chen, et al, “A self-organizing HCMAC neural-network classifier”, IEEE Transactions, Neural Networks, vol. 14, 2003,15–27.

Hybrid Particle Swarm Optimization with BFGS Method

Kezhong Lu
Department of Computer Science, Chizhou Teachers College, Chizhou 247000, China
E-mail: luke76@163.com
Xiaoying Shuai
Department of Computer Science, Chizhou Teachers College, Chizhou 247000, China
E-mail: xyshuai@163.com

ABSTRACT

To overcome the problem of premature convergence on particle swarm optimization (PSO) in optimizing multi-modal function, this paper proposed a hybrid algorithm combining PSO and BFGS method (PSO-BFGS) and used a special mutation to make particles escape local minima. Three benchmark functions were selected as the test functions. The result shows that the hybrid PSO-BFGS algorithm can not only effectively locate the global optimum, but also have a rather high convergence speed. This hybrid algorithm is a promising approach for solving global optimization problems.

Keywords: Particle Swarm Optimization, Hybrid Algorithm, Global Optimum, BFGS Method.

1. INTRODUCTION

Particle swarm optimization (PSO) algorithm, originally introduced by Kennedy and Eberhart in 1995[1, 2], is an evolutionary computation technique, which is inspired by social behavior of swarms. PSO, like the other evolutionary computation algorithms, can be applied to solve most optimization problems: system design, multi-objective optimization, classification, pattern recognition, biological system modeling, etc [3]. The reasons that PSO is attractive are faster convergence speed, very few parameters to adjust and come true easily. But Angeline [4] points out that the PSO performs well in the first phase of the search, and then slows down or even becomes stagnant as the number of iterations is increased. Once the algorithm slows down, it is difficult to achieve better fitness values.

To overcome this problem, several investigations have been undertaken to improve the performance of standard PSO (SPSO). Lobjerg et al.[5] presented a hybrid PSO model with breeding and subpopulations. Shi et al.[6] proposed fuzzy adaptive weighted swarm optimization, which introduced a new update of the weight. X.F. Xie et al.[7] proposed adaptive particle swarm optimization on individual level. In chaotic particle swarm optimization proposed by Jiang and Etorre[8], the objective was to introduce chaos theory so that chaotic mapping enjoys certainty, ergodicity and stochastic property to improve the exploration ability of the algorithm. G.C. Chen and J.S. Yu presented a hybrid PSO model with Simplex Method [9].

In this paper, we propose a hybrid algorithm combining PSO and BFGS method with mutation operator. The BFGS method can enhance the hybrid algorithm's local search ability, the mutation operator can help hybrid algorithm escape from local minima. Three benchmark functions are selected as the test functions. The experimental results show that the hybrid algorithm can not only effectively locate the global optimum, but also have a rather high convergence speed.

2. HYBRID PARTICLE SWARM OPTIMIZATION

2.1 Standard Particle Swarm Optimization(SPSO)

Like other evolutionary algorithms, PSO is a population based search algorithm and is initialized with a population of random solutions, called particles. Each particle flies in the *D*-dimensional problem space with a velocity which is dynamically adjusted according to the flying experiences of its own and its colleagues. The location of the *i*th particle is represented as $X_i=(x_{i1}, x_{i2}, ..., x_{iD})$. At each generation, each particle is updated by following two 'best' values. The first one is the best previous location (*pBest*) a particle has achieved so far. The *pBest* of the *i*th particle is represented as $P_i= (p_{i1}, p_{i1...} p_{iD})$. The second is the best *pBest* among all the particles is represented by P_g (*gBest*.).The velocity for the *i*th particle is represented as $V_i= (v_{i1}, v_{i2}... v_{iD})$.

The particle swarm optimization concept consists of, at each time step, changing the velocity and location of each particle toward its *pBest* and *gBest* locations according to the equations (1) and (2) respectively:

$$v_{id}=w* v_{id} +c_1*\text{rand1}()*(p_{id}- x_{id})+ c_2*\text{rand2}()*(p_{gd}- x_{id}) \quad (1)$$

$$x_{id}= x_{id} +v_{id} \quad (2)$$

where rand1() and rand2() are two random numbers independently generated within the range of [0,1], and c_1 and c_2 are two learning factors which control the influence of the social and cognitive components, *w* is inertial weight which balances global and local search.

In equation (1), if the sum on the right side exceeds a constant value, then the velocity on that dimension is assigned to be $\pm V_{max}$. Thus, particles' velocities are clamped to the range of $[-V_{max} , V_{max}]$ which serves as a constraint to control the global exploration ability of the PSO algorithm. This also reduces the likelihood of particles for leaving the search space. It limits the maximum distance that a particle will move during one iteration [3].

The process for implementing PSO is as follows:

1) Set current iteration generation t=1. Initialize a population which including *M* particles. For the *i*th particle, it has random location X_i in specified space and for the *d*th dimension of V_i, v_{id}=0;

2) Evaluate the fitness for each particle;

3) Compare the evaluated fitness value of each particle with its *pBest*. If current value is better than *pBest*, then set the current location as the *pBest* location. Furthermore, if current value is better than *gBest*, then reset *gBest* to the current index in particle array;

4) Change the velocity and location of the particle according to the equations (1) and (2), respectively;

5) $t=t+1$, loop to step 2 until a stop criterion is met, usually a sufficiently good fitness value *eps* or *t* reaches a predefined maximum generation T_{max}.

2.2 BFGS method[10]

BFGS is a quasi-Newton optimization method that uses information about the gradient of the function at the current point to calculate the best direction to look in to find a better point. Using this information, the BFGS algorithm can iteratively calculate a better approximation to the inverse Hessian matrix, which will lead to a better approximation of the minimum value.

From an initial starting point, the gradient of the function is calculated and then the algorithm uses this information to calculate the best direction to perform a line-search for a point that is "sufficiently better". The algorithm will then adjust the current estimate of the inverse Hessian matrix, and restart from this new point. If a better point cannot be found, then the inverse Hessian matrix is reset and the algorithm restarts from the last accepted point.

The process for implementing BFGS is as follows:

1) Set the variable's dimension n and the desired fitness ε, and initialized starting point X^1, $X^1 \in R^n$;

2) Calculate the gradient of the function at point X^1: $g_1 = \nabla f(X^1)$, if $\| g_1 \| \leqslant \varepsilon$ then set $X^* = X^1$ and $f^* = f(X^1)$, stop the algorithm, else goto step 3;

3) Set current iteration generation k=1 and Hessian matrix $H_1 = I_n$, where I_n is n-identity matrix;

4) Calculate the λ_k:

$\min_{\lambda \geq 0} f(X^k + \lambda Z^k) = f(X^k + \lambda_k Z^k)$, where $Z^k = -H_k g_k$;

5) Set $X^{k+1} = X^k + \lambda_k Z^k$. Calculate $g_{k+1} = \nabla f(X^{k+1})$, if $\| g_{k+1} \| \leqslant \varepsilon$, then set $X^* = X^{k+1}$ and $f^* = f(X^{k+1})$, stop the algorithm, else goto step 6;

6) Set $\triangle X_k = X^{k+1} - X^k$ and $\triangle g_k = g_{k+1} - g_k$. Calculate

$$H_{k+1} = H_k + (1 + \frac{\Delta g_k^T H_k \Delta g_k}{\Delta X_k \Delta g_k}) \frac{\Delta X_k \Delta X_k^T}{\Delta X_k^T \Delta g_k} - \frac{\Delta X_k \Delta g_k^T H_k + H_k \Delta g_k \Delta X_k^T}{\Delta X_k^T \Delta g_k};$$

7) Set $k = k+1$, goto step 4.

2.3 Hybrid particle swarm optimization(HPSO)

PSO is a good global optimization algorithm in the early iterations, but has problems in reaching a near optimal solution; its convergence speed slows down or even becomes stagnant. Once the algorithm slows down, it is difficult to achieve better fitness values. We believe the reasons for this problem are: PSO algorithm hasn't good local search ability near optimal solution; as evolution goes on, PSO algorithm might undergo an undesired process of diversity loss. BFGS method has good ability to local search with a super-linear convergence rate. For enhancing PSO algorithm's local search ability, we introduce BFGS method in the late phase of the search. But BFGS method can't stop the diversity losing, for keeping the particles' diversity, we introduce mutation operator. When mutate? By testing, we discovered if the iteration time was large at calculating the value λ_k on the BFGS method's step 4 that indicated the PSO algorithm had run into local optimal solution. So we set a maximum iteration time Kt to step 4, if iteration has reached Kt, then stop BFGS method and produce new population with mutation operator by equation (3), where C_t is set to a larger positive value, that means the new population is produced at a wide scope and the diversity of population can be kept up; if iteration never reaches Kt, that indicates the BFGS method's optimization direction is positive, then produce new population by equation (3) where C_t is set to a smaller positive value.

$$X^n(k) = (\text{rand}(k) - 0.5) \times C_t \times X^* + X^* \quad (k = 1, \ldots, M) \tag{3}$$

Where X^n is new population, rand(k) is 0~1 random number, M is population size.

Here, we use an approximate method to calculate the value λ_k, namely, iteratively calculate a value λ_k which is satisfied with equations (4) and (5).

$$f(X^k + \lambda_k Z^k) \leq f(X^k) + \alpha \lambda_k \nabla f^T(X^k) Z^k \tag{4}$$

$$\nabla f^T(X^k + \lambda_k Z^k) Z^k \geq \beta \nabla f^T(X^k) Z^k \tag{5}$$

where $\alpha \in (0, 1/2)$, $\beta \in (\alpha, 1)$.

The process for calculating the value λ_k is as follows:

```
begin
  initialize α, β, K_t, k=0, λ_k =1 and bool=1
  while(bool and k<K_t) do
   begin
    if(f(X^k+λ_k Z^k)>f(X^k)+αλ_k ∇f^T(X^k) Z^k) then
      λ_k =0.5*λ_k
    else if( ∇f^T(X^k+λ_k Z^k)Z^k<β ∇f^T(X^k)Z^k) then
      λ_k =0.5*λ_k
    else
      bool=0
    endif
   end
  k=k+1
 endwhile
 if(k>= K_t) then
    set a large positive value to C_t
 else
    set a small positive value to C_t
 endif
end
```

The structure of the hybrid model is as follows:

```
begin
 initialize
 while (not terminate-condition) do
  begin
   execute PSO algorithm until a stop criterion is met
   if (not terminate-condition) then
    execute BFGS method from the initial starting point P_g
achieved by PSO above
   endif
   if (not terminate-condition) then
    produce new population by equation (3)
   endif
  end
 endwhile
end
```

3. EXPERIMENTAL SETTINGS AND RESULTS

Three benchmark problems are tested here. The first function is unimodal, while the last two are multimodal with many local minima. All functions have same minimum value, which are equal to zero.

The first test function f_1 is the generalized Rosenbrock function given by the equation:

$$f_1(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \tag{6}$$

The second function f_2 is the generalized Rastrigin function which is given by the equation:

$$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{7}$$

The final test function f_3 is the generalized Griewank function:

$$f_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1 \quad (8)$$

These three functions have been commonly used in other studies on particle swarm optimizers.

The initial population is usually uniformly distributed over the entire search space. According to [4] this can give false indications of relative performance, especially if the search space is symmetric around the origin where many test functions have their global optimum. To prevent this, and to ease comparison with other models, the asymmetric initialization method used in [4] was used. Search space and initialization ranges for the experiments are listed in table 1.The dimensions D for the three functions are 20. The desired fitness *eps* is 0. In both the standard PSO and the hybrid PSO, the acceleration constants are set as: c_1=c_2=2, the number of particles M is 20 and a linearly decreasing inertia weight starting at 0.9 and ending at 0.4 is used. The maximum velocity V_{max} of each particle is set to be half the length of the search space in one dimension (for instance V_{max} =100 for f_1). The T_{max} in SPSO is equal to 1e4, in hybrid PSO is 1e3. For BFGS method in hybrid PSO, the critical constant ε is set as 1e-16,1e-6 and1e-6 for f_1, f_2 and f_3 respectively; K_t is set to 10, mutation operator C_t is set to 100, and its default value is set to 2; the parameters α and β are 0.1 and 0.7 respectively. The parameters of alone BFGS method are same as above, except that the critical constant ε for f_1 function is set as 1e-6. We had 50 trail runs for every instance.

Table 1. Search space and asymmetric initialization ranges for each test function

Function	Search space	Initialization range
f_1	$-100 \leqslant x_i \leqslant 100$	$15 \leqslant x_i \leqslant 30$
f_2	$-10 \leqslant x_i \leqslant 10$	$2.56 \leqslant x_i \leqslant 5.12$
f_3	$-600 \leqslant x_i \leqslant 600$	$300 \leqslant x_i \leqslant 600$

Table 2. The mean fitness value and function's mean running time for each function and each algorithm

Function	Algorithm	Mean fitness	Function's mean running time
f_1	PSO	1.2053e3	1e4
	BFGS	0.7973	4.5160e6
	HPSO	0	1.7948e4
f_2	PSO	13.2329	1e4
	BFGS	287.6604	1.2202e4
	HPSO	0	5.9657e3
f_3	PSO	122.8488	1e4
	BFGS	5.9129e-4	1.3160e5
	HPSO	0	4.1695e3

Table 2. Shows results for the experiments with standard PSO, BFGS method and hybrid PSO-BFGS respectively. By compare the results, it is easy to see that HPSO have better than all of other algorithms for all cases. For example, as SPSO get worst solutions due to premature convergence for Rosenbrock and Griewank function; BFGS method get worst solution due to being sensitive to selecting initial starting point for Rastrigin function; HPSO can not only achieve global optimal solution 0 for each function and each time running, but also not need too many counting steps. It means that HPSO is a promising approach for solving global optimization problems.

4. CONCLUSION

In this paper, a hybrid PSO-BFGS algorithm with mutation operator is introduced to improve the performance of standard PSO. The hybrid algorithm can not only make use of PSO's global search ability and BFGS method's local search ability, but also can keep the diversity of particle swarm with mutation operator. Three benchmark functions have been used for testing. The simulation results illustrate the hybrid algorithm with mutation operator can improve the performance of standard PSO greatly, at least for the three benchmark functions. Future works shall involve in applying the hybrid algorithm to solve real world problems, like scheduling, network routing and power forecasting.

5. REFERENCES

[1] J. Kennedy, R. Eberhart, "Particle swarm optimization", In: IEEE International Conference on Neural Networks,Perth,Australia,November 1995,1942-1948.

[2] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory". In: Proceeding of the sixth international symposium on Micro Machine and HumanScience, Nagoya, Japan, October 1995,39-43.

[3] R. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources", In: Proceeding of Congress on Evolutionary Computation, Piscataway, NJ, May 2001,81-86.

[4] P.J. Angeline, "Evolutionary optimization versus particle swarm optimization: philosophy and performance differences", Conference on Evolutionary Programming, March 1998, 1956-1959.

[5] M.Lovbjerg, T. Rasmussen, T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations", In: Proceeding of the Genetic and Evolutionary Computation Conference, San Francisco, California, March 2001, 469-476.

[6] Y. Shi, R. Eberhart, "Fuzzy adaptive particle swarm optimization", In: Proceeding of the Congress on Evolutionary Computation, Seoul, Korea, Vol.1, July 2001, 101-106.

[7] X.F. Xie, W.J. Zhang, et al, Yang, "Adaptive particle swarm optimization on individual level", In: International Conference on Signal Processing, Beijing, China, August 2002,1215-1218.

[8] C.W. Jiang, B. Etorre, "A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization", Mathematics and Computers in Simulation, Vol.68, No.1, April 2005, 57-65.

[9] G.C. Chen, J.S. Yu, "Simplex particle swarm optimization algorithm and its application", Journal of System Simulation, Vol.18, No.4, April 2006, 862- 865.

[10] Y.X. Yuan, W.Y. Sun, Optimal theory and method, Beijing: Science Press, Pub., 1997.

Improved VRP Based on Particle Swarm Optimization Algorithm

Zixia Chen
Center for Research in Modern Business Zhejiang Gongshang University
Hangzhou, Zhejiang ,310035, China
Email:czx@mail.zjgsu.edu.cn
And
Youshi Xuan
College of Computer and Information Engineering, Zhejiang Gongshang University
Hangzhou, Zhejiang ,310035, China
E-mail:xuanyoushi123@tom.com

ABSTRACT

The Improved Vehicle Routing Problem (IVRP) and Location Allocation Problems (LAP) were considered synthetically. And a mathematic model about IVRP has been built up. The purpose of this model was to minimize the total cost by determining the location of the warehouse, allocating appropriate number of vehicles for the selected warehouse and finding the optimal routing for each vehicle. At the same time, Improved Particle Swarm Optimization Algorithm with Genetic Algorithm and Simulated Annealing were applied to solve an example of the model. The good convergence state of the global best solution has proved the IVRP model was correct and the algorithm applied to solve IVRP was effective.

Keywords*:* IVRP, Logistics, model, Particle Swarm Optimization Algorithm, Genetic Algorithm, Simulated Annealing.

1 INTRODUCTION

In the times of E-commerce, for many companies in order to satisfy their customers and take a leading edge in the market, the fast、efficient distribution of goods has become more important than ever before. In distribution planning, one key question needs to be taken care of, namely how the customers can be served in an optimal way? While the number of orders that need to be fulfilled steadily increases, most orders nowadays tend to be small and hence delivery cannot be efficiently done on single trip to each customer.

As 3PL (Third Part Logistics) has been adopted by more and more enterprises, the distribution may appear as follows: Vehicles provided by 3PL set out from the enterprises' warehouses; distribute the goods to every customer. After having accomplished the distribution, the vehicles continue to execute another new distribution assignment according to the schedule of 3PL. and when distribution strategies are made, it's not necessary for enterprises to consider such state constraints as whether vehicles have returned the warehouses from where the vehicles set out. So the routes in this situation have turned into Hamiltonian Path from Hamiltonian Cycle. As a result, Open Vehicle Routing Problems (OVRP) with regard to the Hamiltonian Path is attracting more and more scholars. According to the integrated logistics, the integration of OVRP was first described in a paper by Schrage (1981) without any solution attempt. Bodinetal (1983) addressed the express airmail distribution in the UAS and solved two separate routing problems, one for the delivery and another one for the pick-ups using a modified Clarke-Wright savings algorithm. Sariklis and Powell (2000) proposed a classical heuristic method to solve a symmetric OVRP that did not include maximum route length restrictions.

In order to optimize the distribution network with open vehicle routes, the integration of OVRP and Location Allocation Problems (LAP) may give birth to a better logistic scheme (Improved Vehicle Routing Problems, IVRP) which decreases the logistic cost remarkably. IVRP holds a central place in distribution management. It consists of designing a set of least cost vehicle routes at a depot, and visiting a set of customers. Subject to various constraints such as vehicle capacity or route length restrictions. The IVRP is very hard to be solved exactly.

The conception of Location Routing Problems (LRP) was first put forward in 1961. In its most general form, the location-routing problems (LRP) seek to minimize total cost by simultaneously selecting a subset of candidate facilities and constructing a set of delivery routes. Foreign scholars had done many researches in this field. Domestic scholars started to pay attention to it in 2000, and had got some achievements. With the emergence of 3PL, scholars started to focus on OVRP, but there aren't so many researches on it. The research on the model and algorithm of IVRP hasn't been found so far.

Particle Swarm Optimization (PSO) Algorithm was proposed by James Kennedy and R· C· Eberhart in 1995, motivated by social behavior of organisms such as bird flocking and fish schooling. PSO algorithm is not only a tool for optimization, but also a tool for representing sociocognition of human and artificial agents, based on principles of social psychology. Some scientists suggest that knowledge is optimized by social interraction and thinking is not only private but also interpersonal .PSO as an optimization tool, provides a population-based search procedure in which individuals called particles change their position (state) with time. In a PSO system, particles fly around in a multidimensional search space. During flight, each particle adjusts its position according to its own experience, and according to the experience of a neighboring particle, making use of the best position encountered by itself and its neighbor. Thus, as in modern Genetic Algorithms (GA) and memetic algorithms, a PSO system combines local search methods with global search methods, attempting to balance exploration and exploitation. Since this algorithm needs fewer units and could be performed more easily and robustly than other optimization algorithms, it is very fit to solve the multidimensional continuous optimization problems. When it comes to VRP, overseas and domestic scholars have put forward the improved PSO Algorithm.

In this paper, a mathematic model about IVRP has been built up, and an improved Particle Swarm Optimization Algorithm was used to resolve this model.

2 MATHEMATIC MODEL OF IVRP

2.1 Hypotheses

In IVRP, the basic hypotheses are as follows:

1) the demand of every customer and the transportation cost among customers are determinate;
2) the constructing expense of warehouses at every candidate site and the transportation cost between the candidate sites of warehouses and customers are determinate;
3) there are many vehicles that can be used to distribute but the number of them isn't determinate, and the carrying capacity of each vehicle is equal and determinate; it isn't essential for vehicle to return the starting warehouse after distribution task has been completed, instead it could quit this distribution network from the customer where the task is completed;
4) each customer is only supplied by the fine vehicle, and each customer has to accept only one service at one time;
5) the goal of the model is to minimize the total cost that is composed of the constructing expense of warehouse, the usage cost of vehicles and the transportation cost.

2.2 Known conditions

N, *M* respectively represent the total number of customers*(i=1,2,3,...,N)* and candidate sites of warehouses *(i=N+1,N+2,N+3,...,N+M)*; the demand of customers is *Di; (i=1,2,3,..., N)*; the transportation cost between customers is *Cij(i,j =1,2,3,...,N)*;the transportation cost between the candidate sites of warehouse and the customers is *Cij, (i=N+1, N+2, N+3, ..., j=1,2,3,...,N)*; the constructing expense at each candidate site is *Ti(i=N+1,N+2,N+3,..., N+M)*; the carrying capacity of the vehicle is *L*; the usage cost of each vehicle is *C*.

2.3 Decision variables

X_{ij}^{gk} --Binary variable, if the vehicle k is setting out from warehouse *g* to customer *j* by way of customer(or warehouse)i $X_{ij}^{gk}=1$, otherwise, $X_{ij}^{gk}=0$;

Zg--Binary variables, if a warehouse is built at the candidate site *g*, *Zg=1*, otherwise *Zg =0*;

Kg--the number of the vehicles standby in the candidate warehouse g.

2.4 Mathematic mode

Mathematic model is as follows:

$$\text{Min}\sum_{g=N+1}^{N+M}[Z_g(T_g+CK_g)+\sum_{k=1}^{K_g}\sum_{i=1}^{N+M}\sum_{j=1}^{N}Z_g X_{ij}^{gk}C_{ij}] \quad (1)$$

$$S.T$$

$$\sum_{g=N+1}^{N+M}\sum_{k=1}^{K_g}\sum_{i=1}^{N+M}X_{ij}^{gk}=1,\ \forall j\in\{1,2,\ldots,N\} \quad (2)$$

$$\sum_{i=1}^{N+M}\sum_{j=1}^{N}X_{ij}^{gk}Q\le L,\ \forall g\in\{N+1,N+2,..N+M\},\ \forall k\in\{1,2,\ldots,k_g\} \quad (3)$$

$$\sum_{g=N+1}^{N+M}\sum_{i=1}^{N+M}X_{ij}^{gk}=\sum_{g=N+1}^{N+M}\sum_{j=1}^{N+M}X_{jj}^{gk}\forall p\in\{1,2,\ldots,N\},\ \forall k\in\{1,2,\ldots,K_g\} \quad (4)$$

$$\sum_{j=N+1}^{N+M}X_{ji}^{mk}=\sum_{j=N+1}^{N+M}X_{ij}^{gk}=0,\ \forall k\in\{1,2,\ldots,K_i\},\ \forall i=m\in\{N+1,N+2,\ldots,N+M\} \quad (5)$$

$$\sum_{j=1}^{N}X_{gj}^{gk}=Zg,\ \forall k\in\{1,2,\ldots,Kg\}, \quad (6)$$

$$\sum_{k}^{kg}\sum_{i=1}^{N}X_{gi}^{gk}\le K_p,\ \forall g=p\in\{N+1,N+2,\ldots,N+M\} \quad (7)$$

$$X_{ij}^{gk},Z_g\in\{0,1\} \quad (8)$$

Formula(1) is the objective function, which means minimizing the total cost of warehouses、Vehicles and transportation;
Constraint (2) ensures each customer has to be served by only one vehicle from the selected warehouse,;
Constraint (3) restricts the carrying capacity of vehicles;
Constraint (4) shows vehicles are forbidden to stop at the customer;
Constraint (5) shows there isn't any vehicle link between every two selected warehouses;
Constraint (6) shows each vehicle has to set out from a decided warehouse;
Constraint (7) shows the number of the vehicles setting out from the decided warehouse can't exceed the number of the vehicles supplied by 3PL to this warehouse;
Constraint (8) shows the range of decision variables compared with OVRP of single warehouse, the decision variables of the IVRP involve the location variable of warehouse besides, while compared with standard LRP, constraints of the IVRP are more flexible. In the IVRP, vehicles needn't return the warehouse from where they setting out. In this paper, it is assumed that the number of vehicles and their starting locations are determined by the model itself.

Vehicles and their starting locations are determined by the model itself.

3 SOLUTION TO IVRP BASED ON IMPROVED PARTICLES WARM OPTIMIZATION ALGORITHM

In PSO Algorithm, each particle exchanges information with its own historical best position and global best particle of the whole swarm, then looks for the best solution at last. Based on candidate solutions, PSO can make use of particles' own information, unit extremum information and global extremum information to look for the best solution. However, the standard PSO can't effectively solve the problem as the

speed can't be expressed easily if the solution space for the routes expressed by serial number. In the improved PSO, Genetic Algorithm is applied to solve this problem: On one hand, apply mutation process of GA, execute mutation operation on particle itself, look for the best solution based on the particle's own information, on the other hand, apply crossover process of GA, carry out crossover operation on each particle with historical best position of this particle and global best particle, and look for the best solution based on unit extremum information and global extremum information; After executing operation of mutation and crossover, the best or the acceptable solution of particle may be destroyed, so the improved algorithm, based on Simulated Annealing, was put forward, which allows the destroyed range of objective function, aiming at new particle after mutation and crossover operation. According to the method above, the encoding method of the algorithm solving the open location routing problems is as follows:

Each particle is composed of three arrays with the same length which is equal to the number of customers, every dimension of each array is attached to one customer in turn, which respectively stands for the serial number of warehouse that serves this customer, the serial number of vehicle and the order of customer in the route of distribution. This method enables crossover process and mutation process of GA to implement conveniently and effectively on each particle.

At every iteration, each particle in the swarm has to execute mutation operation on itself and crossover operation with its own previous best position and global best particle in the whole swarm. Because too many customers involved during these operations, it may destroy the good or better particle. So single-pot mutation operator and single-pot crossover operation are applied in this algorithm. The basic flow of the improved Particle Swarm Optimization Algorithm applied to solve the IVRP is as follows:

Step1: apply the NNH (Nearest Neighbor Heuristic) method to get initial solution and build up the initial particle swarm;

Step2: compute the value of objective function of each particle, initialize each particle's best historical position and look for global best particle in the whole swarm;

Step3: execute single pot mutation operation on each particle in the swarm;

Step4: execute single pot crossover operation on the new particle after mutation, with its best historical position and globally best particle;

Step5:compute the value of objective function about this new particle obtained in step 4. If the value is lower than the previous generation's, then adopt the new particle, otherwise create a number randomly. If this random number is less than the acceptable probability, then adopt this particle, else refuse to change the particle;

Step6: look for the best historical particle of each particle;

Step7: re-find the best particle in the new swarm. If the result of objective function is better than that computed by global best particle previous, adopt this new global best particle;

Step8: if the iteration times are less than the times to exit, then go to step3 ; else exit.

4 EXAMPLES

The improved algorithm based on the Genetic Algorithm and Simulated Annealing is applied to solve the OVRP. There are 5 candidate sites of warehouses and 25 customers in the example. Such data as the transportation cost, constructing expense are produced by a random generator. The value of objective function of global best solution, which is produced in every iteration, is shown in Fig. 1.

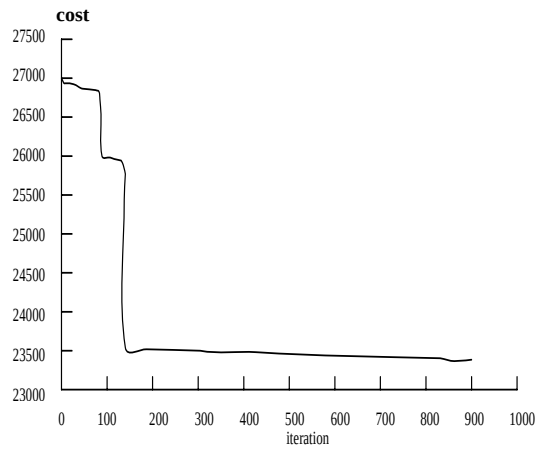


Fig. 1. The value of objective function of global best

As shown in Fig.1,the value of objective function starts to converge after the 804th iteration, and the value of the best solution obtained finally is 234249.891. Meanwhile, the trend of the convergence is notable, that has indicated the IVRP Model is correct and the improved algorithm is suitable to solve this problem. The selected warehouses、 vehicles and the routing, which are determined by the best solution are shown in Table 1.

Table 1. The scheme determined by the best particle

The selected warehouse	The vehicles schedule	The routing
1	1	17→9→4→6
	2	12→1→21
3	3	18→10→7
	4	8→5→11→14
	5	16→24→20
4	6	23→22→13→2
5	7	15→25→3→19

5 CONCLUSION

An mathematic model of the IVRP has been built up. The purpose of this model was to minimize the total cost by determining the location of the warehouse, allocating appropriate number of vehicles for the selected warehouse and finding the optimal routing for each vehicle. At the same

time, Improved Particle Swarm Optimization Algorithm based on Genetic Algorithm and Simulated Annealing were applied to solve an IVRP example, and the notable trend of the convergence has proved that the IVRP Model is correct and the algorithm is suitable.

6 REFERENCES

[1]Z.X. Chen, "A GridNetwork-Dividing Algorithm of the Distribution Line in Hangzhou Tobacco Company", *Journal of Shanghai Jiaotong University*, 2003, 37(7), pp.1013-1017.

[2] Z. Fu, "The capacitated open vehicle routing problem and its tabu search algorithm",Systems Engineering-Theory & Practice, No.3, 2004, pp.123-128.

[3]Jayaraman, R. Srivastava,"Combined location-routing problem: a synthesis and future research directions", European Journal of Operational Research, 1998, 108 (1), pp.1-15.

[4] Ambrosino, Scutella, et al, "Distribution network design: new problems and related models", European Journal of Operational Research, 2005, 165(3), pp.610-624.

[5] Z.X. Chen, Q.T. Ye, "An Algorithm Research on Single Vehicle Routing Problem Based on Real Streets Distribution", Computer Engineering, 2005, 31(11), pp.32-34.

[6] S. Y., Wang, et al, "Research on combined location-routing problems in integrated logistics systems". Journal of Management Sciences in China, 2000, 3(2), pp.69-75.

[7] Q. Zhang, L.Q. Gao, et al, "A two-phase heuristic approach to the location routing problem". Control & Decision, 19(7),2004, pp.773-777.

[8] D. Sariklis, S Powell, "A heuristic method far the open vehicle routing problem". Journal of the Operational Research Society, 51(5),2000, pp.564-573.

[9]J. Grefenstette, R. Gopal, et al, "Genetic algorithms for the traveling salesman problem", Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985, pp. 160-168.

[10] G. Desaulniers, J. Desrosiers, et al, "The VRP with pick up and delivery", Cahiers du GERARD G-2000-25,Ecole des Hautes Etudes Commerciales, Montreal,2000.

[11] A. A. Assad, "Modeling and implentation issues in vehicle routing", B. L. Golden and A. A. Assad, Eds., Vehicle Routing: Methods and Studies, Elsevier Science Publishers, North-Holland, Amsterdan, 1988, pp. 7-45.

[12] A. Archer, A. Levin, et al, "A faster, better approxi-mation algorithm for the minimum latency problem", Technical Report1362, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 2003.

[13] Thangiah SR, Nygard PL, et al, "GIDEON: a genetic algorithm system for vehicle routing with time windows", Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications, Miami, FL, 1991, pp. 322-328.

[14] Salhi S, Thangiah SR, et al, " A genetic clustering method for the multi-depot vehicle routing problem.", Smith GD, Steel NC, Albrecht RF, editors. ICANNGA'97, Vienna. New York, Springer, 1998, pp. 234-237.

[15] P. Chalasani, R. Motwani, et al, "Algorithms for robot grasp and delivery", Proceedings of the 2nd Int. Workshop on Algorithmic Foundations of Robotics (WAFR), 1996, pp.347-362.

Several Critical Techniques of Genetic Programming and Their Applications for Data Mining

Yongqiang Zhang
School of Information and Electricity-Engineering, Hebei University of Engineering
Handan, Hebei Province, P.R.China
Email: yqzhang@hebeu.edu.cn

And

Huashan Chen
School of Information and Electricity-Engineering, Hebei University of Engineering
Handan, Hebei Province, P.R.China
Email: chs811013@163.com

ABSTRACT

A common problem in data mining is to find accurate data fitting and trend-based forecasting for a dataset. Genetic Programming (GP for short) was accordingly applied, which can particularly induce parse trees with a linear combination of variables in each function node. Different methods of selection, crossover and mutation were also adopted which can be used to avoid the undesirable growth of program size. Additionally, ordinary differential equations and the Particle Swarm Optimization (PSO for short) were used to improve the accuracy of data fitting and forecasting. The results indicate that the improved GP approaches can be applied successfully for accurate data fitting and forecasting.

Keywords: GP, program size, ordinary differential equation, PSO, data mining.

1. INTRODUCTION

GP is a domain independent problem solving approach in which computer programs are evolved to solve, or approximately solve problems. GP needs a good random tree-creation algorithm to create trees that form the initial population, and to create sub trees used in sub tree crossover and mutation, which are GP operators used for modifying trees in the population.

Its search space is potentially unlimited and programs may grow in size during the evolutionary process. Unfortunately, it also permits the appearance of pieces of redundant code, which increase the size of programs without improving their fitness. Besides consuming precious time in an already computationally intensive process, it may start growing rapidly, a situation known as bloat. A combination of factors may be involved, but whatever the reasons may be, the fact remains that bloat is a serious problem that may stagnate the evolution, preventing the algorithm from finding better solution [1, 2, 6].

In this view, several different methods of selection, crossover and mutation towards different conditions are adopted, which is closely related to the depth of the tree and the fitness of the solution in one generation. In order to certify the techniques here, we apply them into a dataset of data mining, which can get one better model. The results show that GP can be applied successfully data fitting and forecasting problems.

*Project 60573088 supported by National Natural Science Foundation of China (NSFC);
Project 603407 supported by Natural Science Foundation of Hebei Province.

2. BASIC TECHNIQUES OF GP

The most obviously natural way of creating a theory for GP has been to define a concept of schema for parse trees and to extend Holland's schema theorem to GP to see how schemata would propagate generation after generation under the effects of selection, crossover and mutation.

According to Koza's definition, a schema (parse tree) could be represented as a set of s-expressions, and then let F and T be the function set and the terminal set used in a GP run, respectively. The several techniques of GP will be related next, which can be useful for bloat of the tree and get the better solution.

2.1 Selection

As in common, we must select the best individual of the generation. Generally, only one selection method is chosen to get the best individual of the program solution. But this way can not be fitting for every condition which may be dealt with. Therefore, we put all of the selection methods together, including the roulette wheel selection, total random selection and the tournament selection. We set a parameter as "tsels", which controls different conditions we face to, and assume three conditions as follows:

If tsels equals 0, then we choose the roulette wheel selection; If tsels equals 1, then we choose the total random selection; If tsels equals 2, then we choose the tournament selection.

The evolutionary algorithm of our GP system applies tournament selection and puts the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament. The loser of each tournament is replaced by a copy of the winner. In such a steady state EA the population size is always constant and determines the number of individuals created in one generation.

2.2 Crossover

One-point crossover can affect the propagation of schemata by disrupting some of them, ignoring others or creating some new schemata. One-Point Crossover is a common crossover point is selected in both parents and the one-point crossover works by selecting a common crossover point in the programs and then swapping the corresponding sub trees like standard crossover. In order to account for the possible structural differences between the two parents, one-point

crossover involves two phases. It should be noted that this form of crossover does not increase the depth of the offspring beyond that of the parents. Obviously, this means that some care is needed when selecting the maximum tree depth to be used in the creation of the initial population. This feature is useful to avoid the typical undesirable growth of program size, which can easily lead to over fitting [1, 3, 4, 5].

Meanwhile, the two-point crossover is also used. When the parameter "mode", which controls what crossover methods we choose, equals 1, we choose one-point crossover; otherwise, we may choose two-point crossover.

2.3 Mutation

Point mutation: consists of changing the label of a node into a different one. We assume that it is applied to the nodes of the programs created using selection and crossover as described in the previous sections with probability pm per node. The probability that each node is not altered is 1-pm. If mutation, likes GA, is applied with a probability pm<<1 [1].

2.4 Model evaluation and Fitness function

Throughout this paper, fitness F of an individual program p is calculated as the mean square error (MSE: defined by $mse=(1/n)*E'*diag(Q)*E$) between the predicted output o_{ij}^{pred} and the desired output o_{ij}^{des} for all n training examples and m outputs:

$$F(p)=\frac{1}{n\cdot m}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(o_{ij}^{pred}-o_{ij}^{des}\right)^2+\frac{w}{n}CE=MSE+w\cdot MCE$$

Additionally, a classification error (CE) is computed as the number of incorrectly classified examples. The mean classification error (MCE) is added to the fitness while its influence is controlled by a weight parameter w. In this way, the classification performance of a program determines selection more directly while the MSE term still allows continuous fitness improvements [3]. We put forward the penalty function as:

$$penalty=1/(1+\exp(a_1*(L-a_2)))$$

We use the fitness as follows:

$$fitness=penalty*corrcoef\left(\sqrt{diag(Q)}*Y,\sqrt{diag(Q)}*Y_m\right)$$

in this case, a_1、a_2 are the two parameters in the penalty function, L is defined by the size of the parse tree or the number of the nodes, $corrcoef$ is the correlation coefficient about Y (the real output) and Y_m (the estimation output) [7,8].

3. IMPROVED APPROACH FOR GP

3.1 Ordinary differential equations in GP

In order to solve the problem of modeling and predicting for data series, the ameliorated model is put forward, which is included the ordinary differential equation based on the traditional GP model that has some shortages on the precision and the extrapolated capability of the evolvement. Consequently, the dynamic description capability of the model can be improved. Meanwhile, the complexity of the model can be reformed, and the true data series can be reflected accurately by this model.

3.2 Particle Swarm Optimization for GP

The Particle Swarm Optimization (PSO for short) was originally designed by Kennedy and Eberhart and has been compared to genetic algorithm for efficiently finding optimal or near-optimal solutions in large search spaces. The technique involves simulating social behavior among individuals (particles) "flying" through a multidimensional search space, each particle representing a single intersection of all search dimensions. The particles evaluate their positions relative to a goal (fitness) at every iteration, and particles in a local neighborhood share memories to adjust their own velocities, and thus subsequent positions.

The original PSO formulae define each particle as a potential solution to a problem in D-dimensional space, with particle i represented $X_i=(x_{i1},x_{i2},\cdots,x_{iD})$. Each particle also maintains a memory of its previous best position, $P_i=(p_{i1},p_{i2},\cdots,p_{iD})$, and a velocity along each dimension, represented as $V_i=(v_{i1},v_{i2},\cdots,v_{iD})$. At each iteration, the P vector of the particle with the best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension, and that velocity is then used to compute a new position for the particle. The portion of the adjustment to the velocity influenced by the individual's previous best position (P) is considered the cognition component, and the portion influenced by the best in the neighborhood is the social component.

With the addition of the inertia factor, ω, by Shi and Eberhart, these formulae are:

$$v_{id}=\omega*v_{id}+\eta_1*rand(\ )*(p_{id}-x_{id})+\eta_2*rand(\ )*(p_{gd}-x_{id})$$
$$x_{id}=x_{id}+v_{id}$$

Constants η_1 and η_2 determine the relative influence of the social and cognitive components, and are usually both set the same to give each component equal weight as the cognitive and social learning rate [9].

In that the models created by GP can not be accurate to the data series, the coefficient parameters should be adjusted to fit and forecast the true data mining much more precise. The PSO approach is presented here, which is used to modify the parameters in created GP models.

In order to simplify how to make use of this optimization method, we just take a simple example. Here, if the created GP model is given by the expression $f(x)$, and the new model which has been optimized by PSO can be shown as $g(x)=a\times f(x)+b$. In this view, we can make use of PSO to optimize the two parameters a and b, which can make GP model more precise for data fitting and forecasting.

Examples for these techniques' application above which are given in section 4 as follows.

4. EXPERIMENTS ON THESE TECHNIQUES

In order to indicate these techniques above, we have taken the example of Gas content data series [10]. A group of the true detective data (10 of them) are shown in Table 1. As follows, this describes the variance of gas content at the specific time.

Table 1. Data Series of Gas Content

x	1	2	3	4	5
y	0.25	0.35	0.361	1.051	1.361
x	6	7	8	9	10
y	1.402	2.305	3.0702	3.5892	3.9059

Where, x represents the time series for gas content forecasting, and y stands for the corresponding gas content data series, which means the gas content in the air. Then the GP models, which all make use of the basic techniques of GP to improve the program size, are created to fit for y data series. We have done programming in MATLAB6.5 and then get two solutions to this problem, which are given as follows:

$$f(x) = 0.1 \times x \times \sqrt{x + \ln x - \cos x} \tag{1}$$

$$f(x) = 0.1 \times \left(x \cdot \ln\left(\frac{x^{\ln(x+\sin x)}}{\sqrt{x - \cos\left(x^{\sin x}\right)}} \right) + \sin\left(\frac{\left(x + \sin x - e^{(x+\ln x)}\right) \cdot \ln x}{\sin\left(\frac{x^{\sqrt{x}}}{\sqrt{x}}\right) + \sin x} \right) + \sin x \right) \tag{2}$$

Based on the two models above, we have created the models with ordinary differential equations as well as models optimized by PSO algorithm, which will be described separately as follows:

First, the two GP models with ordinary differential equations are given below, and the first model with ordinary differential of Eq. (1) is shown Eq. (3), the second is shown in Eq. (4). Both of these models' simulation results by MATLAB6.5 are shown in Fig. 1. And Fig. 2. As follows.

$$\begin{aligned} dy &= -4.638023 \cdot u(k-2) + 0.177579 \cdot y(k-1) + 4.631295 \cdot u(k-1) - 0.917151 \\ y(k) &= y(k-1) + dy \end{aligned} \tag{3}$$

$$\begin{aligned} dy &= 5.980528 \cdot y(k-1) - 15.295119 \cdot y(k-2) + 15.954469 \\ y(k) &= y(k-1) + dy \end{aligned} \tag{4}$$

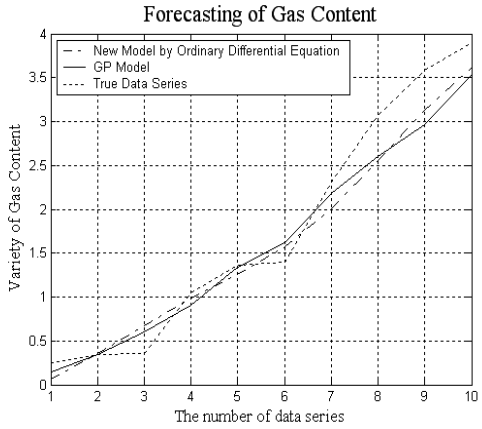


Fig. 1. Simulation of GP model 1 with ordinary differential

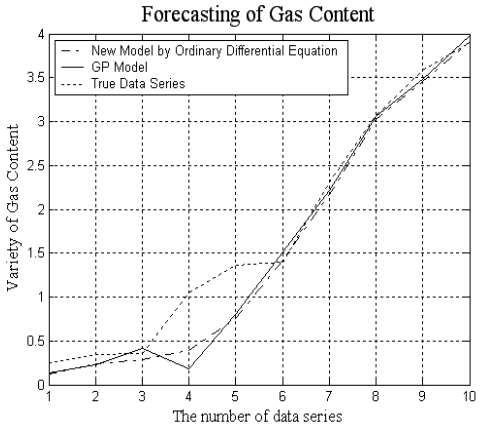


Fig. 2. Simulation of GP model 2 with ordinary differential

In order to testify the feasibility and availability of these models, some of the criteria should be calculated here, such as the correlation coefficient, covariance, mean value and mean square error, the results of these two models created by GP with ordinary differential equations are shown in Table 2. And Table 3. As follows:

Table 2. The difference between GP model 1 with the ordinary differential equation and the original model

Criteria	Correlation coefficient	Covariance
Original model (1)	0.9894	1.3543
New model (1)	0.9892	1.4198
Criteria	Mean value	Mean square error
Original model (1)	1.6258	0.8927
New model (1)	1.6225	0.8171

Table 3. The difference between GP model 2 with the ordinary differential equation and the original model

Criteria	Correlation coefficient	Covariance
Original model (2)	0.9792	2.2063
New model (2)	0.9877	1.3587
Criteria	Mean value	Mean square error
Original model (2)	1.6019	1.1425
New model (2)	1.6430	0.8902

Second, the Particle Swarm Optimization algorithm is adopted to improve the precision of the two original GP models above. Eq. (5) and Eq. (6) have shown the new models of GP with PSO algorithm, their simulation results are given in Fig. 3. and Fig. 4. , and the comparison between the new GP models with PSO algorithm and the original models have presented in Table 4. and Table 5.

$$f(x) = 0.11640 \cdot x \cdot \sqrt{x + \ln x - \cos x} - 0.1478 \tag{5}$$

$$f(x)=0.09298\times\left(x\cdot\ln\left(\frac{x^{\ln(x+\sin x)}}{\sqrt{x-\cos\left(x^{\sin x}\right)}}\right)+\sin\left(\frac{\left(x+\sin x-e^{\left(x+\ln x\right)}\right)\cdot\ln x}{\sin\left(\frac{x^{\sqrt{x}}}{\sqrt{x}}\right)+\sin x}\right)+\sin x\right)+0.2979 \quad (6)$$

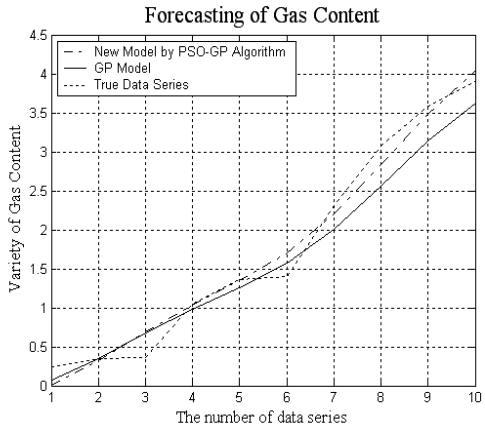


Fig. 3. Simulation of GP model 1 with PSO

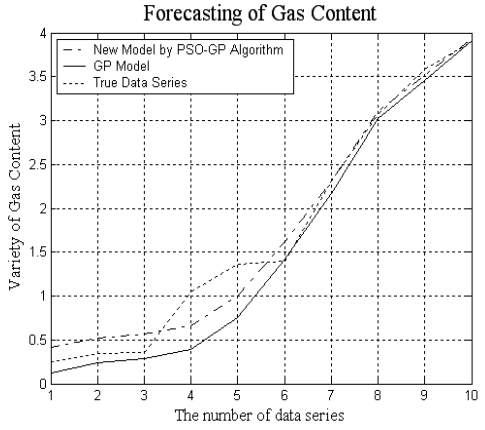


Fig. 4. Simulation of GP model 2 with PSO

Table 4. The difference between GP model 1 with PSO and the original model

Criteria	Correlation coefficient	Covariance
Original model (1)	0.9894	1.3543
New model (1)	0.9895	1.8396
Criteria	Mean value	Mean square error
Original model (1)	1.6258	0.8927
New model (1)	1.7661	0.3553

Table 5. The difference between GP model 2 with PSO and the original model

Criteria	Correlation coefficient	Covariance
Original model (2)	0.9792	2.2063
New model (2)	0.9870	1.8379
Criteria	Mean value	Mean square error
Original model (2)	1.6019	1.1425
New model (2)	1.7645	0.4382

From the comparison of the four parameters above, we know that the two models created by the GP algorithm with ordinary differential equations can get better solutions about our programs than GP itself. Especially from the parameter correlation coefficient, we can see that the new models may be the better ones.

As the correlation coefficient is a quantity that gives the quality of a least squares fitting to the original data. To define the correlation coefficient, consider the value ss_{xy}, which is given in Eq. (7), of a set of n data points (x_i, y_i) about their respective means:

$$\begin{aligned} ss_{xy} &\equiv \sum (x_i - \overline{x})(y_i - \overline{y}) \\ &= \sum (x_i y_i - \overline{x} y_i - x_i \overline{y} + \overline{x}\,\overline{y}) \\ &= \sum xy - n\,\overline{x}\,\overline{y} - n\,\overline{x}\,\overline{y} + n\,\overline{x}\,\overline{y} \\ &= \sum xy - n\,\overline{x}\,\overline{y} \end{aligned} \quad (7)$$

$$r \equiv \sqrt{bb'} = \frac{n\sum xy - \sum x \sum y}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right]\left[n\sum y^2 - \left(\sum y\right)^2\right]}} \quad (8)$$

The correlation coefficient r is then defined by Eq. (8), which is a concept from statistics, is a measure of how well trends in the predicted values. It is a measure of how well the predicted values from a forecast model "fit" with the real data. The correlation coefficient is a number between 0 and 1. If there is no relationship between the predicted values and the actual values the correlation coefficient is 0 or very low (the predicted values are no better than random numbers). As the strength of the relationship between the predicted values and actual values increases, so does the correlation coefficient. A perfect fit gives a coefficient of 1.0. Thus the higher the correlation coefficient, the better model fits.

From the deduction of these techniques on theory and the experiments on these approaches for GP algorithm improving, we can draw conclusions that the improved GP algorithm can be used more precise models, especially for the improved GP algorithm with PSO, which are used for data fitting and forecasting to some extent.

5. CONCLUSION

Several techniques for GP modeling were taken here, including different selection, crossover and mutation, which may be used to avoid the typical undesirable growth of program size. Then we take a dataset of data mining for example using GP ordinary differential equation, which verifies that the results of these techniques in data mining for data fitting and trend-based forecasting can be applied successfully. But to run the GP programs, it will cost more time which affects our efficiency. So the next works we must consider is to use the wavelet theory to decrease the model noise, which is of great worth to be studied further.

6. REFERENCES

[1] R. Poli, A new schema theory for Genetic Programming with One-point crossover and Point mutation. School of Computer Science. 2000.

[2] S. Silva1, J. Almeida, "Dynamic Maximum

[3] Tree Depth-a Simple Technique for avoiding Bloat in Tree-Based GP.Biomathematics Group". Instituto de Tecnologia Qu′ımica e Biol′ogica Universidad Nova de Lisboa, PO Box 127, Oeiras, Portugal. 2002,2780-156.
[4] S. Legg, M. Hutter, Tournament versus Fitness Uniform Selection. Technical Report.2004 (3.20).
[5] J.R.Koza, Genetic Programming. Encyclopedia of Computer Science and Technology. 1997(8.18) 2-4.
[6] K. Stoffel, L. Spector, High-Performance, Parallel, Stack-Based Genetic Programming. Proceeding of the First Annual Conference.1996, 224-229.
[7] S. Luke, L. Panait, Fighting Bloat with Nonparametric Parsimony Pressure. Proceeding of the First Annual Conference.2000.
[8] M. Bot, Application of Genetic Programming to Induction of Linear Classification Trees. Technical Report .1999(11.10).
[9] Y.L. Cao, H.Y. Li, Application of Genetic Programming in Predicting Infinite Dilution Activity Coefficients of Organic Compounds in Water. Chinese Chemical Letters.2003 (Vol.14).
[10] A. Carlisle, G. Dozier, Adapting Particle Swarm Optimization to Dynamic Environments. Proc of Int'1 Conf on Artificial Intelligence. 2000,429-434.
[11] C.F. Wu, Y. Zeng, et al, The application of the analytic methods of neural networks to gas-forecasting field. Advances in earth science. 2004, 10. p.862.

A New Online Scheduling Algorithm for Tasks with (m,k) Deadlines in Overloaded Real-Time Systems*

Xue-Lian Bin1 Yu-Hai Yang1 Ya Bin1 Shiyao Jin2
1 Airforce Army Radar Academy, Wuhan, Hubei 430019,China;
2 National Laboratory for Parallel and Distributed Processing, ChangSha, Hunan 410073, China
Email: Binxuelian@yeah.net

ABSTRACT

(m, k) model is an effective method to reduce the overloads of real-time systems. A new online scheduling algorithm TMS (Triple-Model Schedule) is proposed for scheduling tasks with (m, k) deadlines in overloaded real-time systems. TMS has combined the states and deadlines of requests seamlessly to schedule them. The requests in the critical states are scheduled firstly, thereby avoiding any dynamic failure occurring. In order to avoid requests transferring from quasi-critical states to critical states, we assign their priorities the next highest. The priorities of requests in normal states are the lowest. Finally, the simulation results show that the dynamic failure ratio of TMS is much lower than that of BMS, DWCS, DSP and EDF.

Keywords: Real-Time, Online Scheduling, (m, k) deadline, Breakdown Utilization, Dynamic Failure.

1. INTRODUCTION

In hard real-time systems it is of outmost importance that all deadlines are met. A missed deadline results in a system failure. Since no lateness is allowed, there exists an implicit requirement that worst-case execution time and worst-case arrival rates must be used when scheduling tasks with critical time constraints. In firm real-time systems deadlines should be met but occasional violations are acceptable and do not lead to any damage in the environment. A result that is produced too late, i.e., after its deadline, is however of no value to the system. Consider, for example, the real-time transmission of digitized full motion video. A source (e.g., a video camera) generates a stream of video frames at a rate of, say, 30 frames per second. These frames are transmitted and are played back as they arrive at the destination. Each frame has a deadline before which it must reach the destination. A frame that misses its deadline is dropped and is considered lost. In this application, one can tolerate a few missed deadlines (i.e., dropped frames) without a significant degradation in the video quality. Several models have been proposed to study such systems, e.g., the imprecise computation model, the "skip-over" model, and the (m, k) model. In the (m, k) model, a dynamic failure occurs if less than m out of any k consecutive requests of some tasks meets their deadlines. If m=k, the system becomes a hard-deadline system. For the special case of m=k-1, the (m, k) model reduces to the "skip-over" model. The (m, k) model can be incorporated into Quality of Service (QoS) requirements, and is applicable to many real-time systems such as those in multimedia and automotive control. In this paper, we use the (m, k) model to study the scheduling problem of real-time systems.

Many algorithms of priority assignment and how to abort requests for tasks with (m, k) deadlines have been studied in [1~6, 9~10]. They are mainly divided into two classes: offline and online algorithms. In this paper we focus on the online algorithms. The basic idea of DSP (Distance Based Priority) [1] is to assign higher priorities to the requests of a periodic task that is closer to a dynamic failure so as to improve their chances of meeting their deadlines. Because DSP does not use deadlines of tasks to schedule them, it can't guarantee that each task satisfies its (m, k) constraint. In [6], (m, k) deadlines are guaranteed over fixed windows when DWCS (Dynamic Window-Constrained Scheduling) is used. DWCS is used very restrictively, as it requires the window of a task is not sliding. Therefore, dynamic failures may occur if consecutive (sliding) windows are considered. In [4,5], BMS (Bi-Model Scheduler) is proposed for the tasks with (m, k) deadlines. In the normal mode tasks are scheduled according to a generic scheduling policy, while in the panic mode, in order to avoid a dynamic failure tasks are scheduled according to their static priorities, which are assigned by an optimal priority assignment algorithm [4,5]. This algorithm requires that the utilization factor of a task set be less than or equal to 1 and there is not an existing optimal priority assignment algorithm for a task set with the utilization factor higher than 1. Moreover, the paper [5] points that the time complexity of this optimal priority assignment algorithm is O (n!). Hence, BMS may not be applied to dynamic overloaded real-time systems.

In order to overcome the shortcomings of the above algorithms, we propose a new online scheduling algorithm (TMS) for tasks with (m, k) deadlines in the dynamic overloaded real-time systems.

In section 2, we formally introduce the system model and the problem formulation. After that, in section 3 we present our algorithm. In section 4, we present the performance of the approach conclusions in section 5.

2. SYSTEM MODEL and PROBLEM FORMULATION

In this paper, we consider a periodic tasks set $\Gamma = \{\tau_i \mid 1 \le i \le n\}$ on a single processor. A task τ_i in set Γ is characterized by the parameters: a period T_i, an execution time C_i, a deadline D_i and a constraint condition (m_i,k_i). D_i is arbitrary, that is D_i can be larger than, smaller than or equal to T_i. The offsets in the task model are arbitrarily phased. The tasks are independent (e.g. they do not interact). The j^{th}

* This work was supported by the National Science Foundation of China (NSFC) under the grant No 60073003.

request of a task τ_i is denoted by τ_{ij}. Constraint condition (m_i,k_i) mandates that at least m_i out of any k_i consecutive requests of τ_i must meet their deadlines to avoid any dynamic failure. Every request of a task can be red or blue. A red request of a task must finish its execution before its deadline. Otherwise a dynamic failure will occur. A blue request of a task can be aborted at any time. Hence, a blue request missing its deadline will not cause any damage in the system.

Definition 1

(μ-pattern) A μ-pattern Π of a task is a sequence of symbols of {0,1} that characterizes whether its requests met their deadlines or not. $|\Pi|=p$ is the length of the μ-pattern. $\Pi(k) \in (0,1)$, $1 \le k \le p$. A 1 means that a request of a task has met its deadline and a 0 means that a request of a task has missed it. $\Pi(1)$ characterizes whether the first request of the task met its deadline and $\Pi(p)$ characterizes whether the p^{th} request of the task met its deadline. The sub-string of Π is called sub-μ-pattern, denoted by α. For convenience, the k^{th} element of a sub-μ-pattern α is denoted by α_k.

Definition 2

If a sub-strings α of Π, whose length is k, is satisfied the equation $\sum_{j=1}^{k} \alpha_j = m$, we call α the minimal sub-μ-pattern.

Definition 3

A μ-pattern Π given by figure 1(a) is called the critical μ-pattern. A minimal sub-μ-pattern given by figure 1(b) is called the critical sub-μ-pattern.

$$\overbrace{1\,1\,..\,1}^{m}\ \overbrace{0\,0\,..\,0}^{k-m}\ \overbrace{1\,1\,..\,1}^{m}\ \overbrace{0\,0\,..\,0}^{k-m}\ldots\ .$$

Fig. 1(a) Critical μ-pattern

$$\overbrace{1\,1\,..\,1}^{m}\ \overbrace{0\,0\,..\,0}^{k-m}$$

Fig. 1.(b) Critical sub-μ-pattern

Definition 4

If the current request of a task missed its deadline, a dynamic failure will occur. Then we call the state of the task is critical. The state of a task is called quasi-critical, if its current request missed its deadline then its next request must meet its deadline in order to satisfy its (m,k) constraint. For other cases, the state of a task is normal.

The following will illustrate how to judge the state of a task. Let the sub-μ-pattern β characterize the executions of the most recently consecutive k requests of a task τ_i ($\tau_i \in \Gamma$). According to definition 4, we can draw the following conclusions. If $\sum_{p=1}^{k} \beta_p = m$ and $\beta_1=1$, τ_i is in the critical state. If $\sum_{p=1}^{k} \beta_p = m$ and $\beta_1\beta_2=01$, or $\sum_{p=1}^{k} \beta_p = m+1$ and $\beta_1\beta_2=11$, then τ_i is in the quasi-state. If $\sum_{p=1}^{k} \beta_p = m+1$ and $\beta_1\beta_2 \neq 11$, or $\sum_{p=1}^{k} \beta_p > m+1$, or $\sum_{p=1}^{k} \beta_p = m$ and $\beta_1 \neq 1$ and $\beta_1\beta_2 \neq 01$, then τ_i is in the normal state. If $\sum_{p=1}^{k} \beta_p < m$, a dynamic failure occurs. For this case, τ_i is considered to be in the critical state.

We assume initial states of all tasks are normal and initially all elements in β are 1.

Definition 5

W denotes the window constraint of a task. $W=x/y$. The detail descriptions for x and y are in the [6].

A request of a task inherits the state and the window constraint of the task.

Fig. 2 shows the state transition diagram for a task with (2,3) deadline. The letters C, QC, N represent the critical state, the quasi-critical state and the normal state, respectively. 1 and 0 on the line represent a meet and a miss, respectively. The three numbers in a circle are the elements of a sub-μ-pattern. Lines with arrows represent the possible state transitions. Starting from a state, the task makes a transition to one of the three states, depending on whether its current request met or missed its deadline.

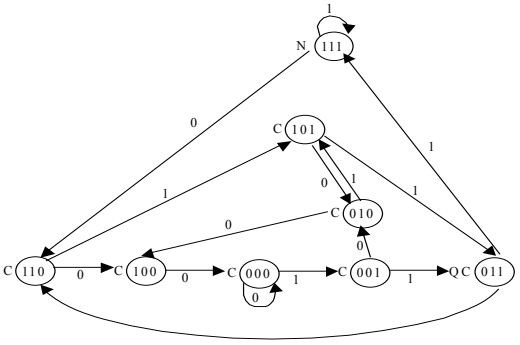


Fig. 2. State transition diagram

3. ONLINE SCHEDULING ALGORITHM TMS

In this section we will develop a new online scheduling algorithm for the tasks with (m,k) deadlines in the overloaded real-time systems.

We know that in the critical states the priorities of tasks scheduled by BMS are fixed. If the priorities of tasks can be successfully assigned off-line, BMS will never produce any dynamic failure. The utilization factor of a set of tasks scheduled by BMS is not very high, because the priorities of tasks in the critical states are fixed. For example, we increase the execution times of tasks in the table 1 gradually at the speed of 1% until the execution time of each task increased to 117%. Table 2 shows the characteristic parameters of tasks with 117% increased execution times. Whichever the priorities of tasks in the critical states are assigned, BMS can't schedule the task set in the table 2 without dynamic failures. In contrast, DSP is able to schedule it without any dynamic failure.

Though DSP is able to schedule the set in table 2 without any dynamic failure, it does not use the deadlines of tasks to schedule them. Hence, high dynamic failures occur. For example, the task set in table 3 can be scheduled by BMS and DWCS without any dynamic failure, but high dynamic failures will occur if it is scheduled by DSP.

Table 1. Task set

τ_i	T_i	C_i	D_i	m_i	k_i	Priority
τ_1	45	22	45	2	4	1(highest)
τ_2	70	22	70	4	6	2
τ_3	225	54	225	1	1	3
τ_4	1200	198	1200	1	1	4(lowest)

Table 2. The set of tasks with 117% increased execution times

τ_i	T_i	C_i'	D_i	m_i	k_i	Priority
τ_1	45	25.74	45	2	4	1(highest)
τ_2	70	25.74	70	4	6	2
τ_3	225	63.18	225	1	1	3
τ_4	1200	231.66	1200	1	1	4(lowest)

Table 3. Task set

τ_i	T_i	C_i	D_i	m_i	k_i
τ_1	1	1	1	1	2
τ_2	1	1	1	1	4
τ_3	1	1	1	2	8

Hence, a good online scheduling algorithm should use both states and deadlines of requests to assign their priorities to reduce dynamic failures. In this paper, we propose a new algorithm TMS, combining the states and deadlines of requests seamlessly.

The following is the basic idea of TMS.

(1) Requests are scheduled in the order of their states: the critical state, the quasi-critical state, and the normal state.

The priorities of requests in the critical states should be the highest to satisfy their (m,k) constraints. In order to avoid requests being switched to the critical states, the priorities of requests in the quasi-critical states should be higher than that of requests in the normal states, lower than that of requests in the critical states. The priorities of requests in the normal states are the lowest because missing their deadlines will not change their states.

(2) Rules of breaking the ties of requests in the critical states (or quasi-critical states) are listed in table 4.

Table 4. Rules of breaking the ties of requests in the critical states (or quasi-critical states)

Earliest Deadline First (EDF)
Equal deadlines, use DWCS to break the ties of the requests in the critical states (or the quasi-critical states). That is, equal deadlines, order the requests with lowest W first; equal deadlines and zero window-constraints, order the highest window-denominator first; equal deadlines and equal non-zero window-constraints, order lowest window-numerator first.
All other cases: first-come-first-serve (FCFS)

(3) Rules of breaking the ties of requests in the normal states are listed in table 5.

Table 5. Rules of breaking the ties of requests in the normal states

EDF
Equal deadlines, order the requests with maximal n first, where n denotes the number of the most recently consecutive k requests that missed their deadlines
All other cases: FCFS

(4) Whenever a request of a task meets its deadline or is aborted due to having missed its deadline, updates its μ-pattern and its state according to definition 1 and 4.

Because TMS is developed to reap the benefits of both DSP and BMS, the dynamic failures of TMS are much lower than the other algorithms.

4. PERFORMANCE COMPARISON

We use randomly generated periodic task sets for our simulations. According to the utilization factors of task sets, we divide them into 12 classes. The utilization factors of these classes are 0.8~0.9, 0.9~1.0, 1.0~1.1, 1.1~1.2, 1.2~1.3, 1.3~1.4, 1.4~1.5, 1.5~1.6, 1.7~1.8, 1.8~1.9, 1.9~2.0, respectively. Each class has 50 sets.

Experiment 1.

This experiment used the same condition as the paper [3] used to generate the task sets that satisfy the following conditions.

1. The utilization factor U of a set is generated in the range of its classes with uniform probability distribution function.
2. Each set has 20 tasks.
3. Each task is characterized by its execution time C, its period T, its deadline D and its (m,k) constraint. T, k and m of a task are generated in the range [10,500], [2,20] and [1,k] with uniform probability distribution function, respectively. The deadline of a task is equal to or less than its period and its execution time follows an exponential distribution with mean equal to $U/20T$.

Experiment 2.

This one uses the same way as paper [14] used to generate task sets. The differences between this experiment from the former one are the methods to generate the execution time and the deadline of a task. In this experiment, the execution time C and the deadline D of a task are generated in the range [0.004,0.5] and [1/2*period, 3/2*period] with uniform probability distribution function, respectively.

BMS use the optimal priority assignment algorithm [4]. If all tasks in a set can be successfully assigned priorities, then the set can be scheduled by BMS without any dynamic failure. However, the optimal priority assignment algorithm just applies to the task sets with utilization factors less than or equal to 1. Therefore, it is a necessity of a quantitative measurement to compare the performance of BMS and TMS. One possibility is to use the binary measure of schedulability on the condition of the utilization factors of task sets are less than or equal to 1, i.e., whether a task set is schedulable with dynamic failures or not. A better measure, and one that is commonly used is the notion of the breakdown utilization [8]. In this paper we use the breakdown utilization to measure schedulability. Here, the breakdown utilization of a task set is defined as the associated utilization factor of a set when the execution time of each task in the task set is scaled by a factor

which results in at least one task violating its (m,k) constraint.

To compare the performance of TMS, DSP, EDF and DWCS, quantitative measurement is much simpler. We use the dynamic failure ratio as the quantitative measurement, which denotes that the number of dynamic failures of all tasks compares with the number of the invocations of all tasks.

Because the priority assignment algorithm used by BMS requires the utilization factor of a task set be less than or equal to 1, we only compare the breakdown utilization of task sets with utilization factors in the range [0.8,0.9] and [0.9,1.0].

The simulation results of the experiment 1 and experiment 2 are shown in table 7, fig. 3, table 8 and fig 4, respectively.

We can see from table 7 and table 8 that the breakdown utilizations of sets scheduled by TMS are much higher than that of sets scheduled by BMS. In the experiment 1 the breakdown utilizations of sets scheduled by TMS are averagely 41.99% higher than that of them scheduled by BMS. In the experiment 2 the breakdown utilization of sets scheduled by TMS are averagely 31.77% higher than that of them scheduled by BMS.

From the fig. 3 and fig. 4, we can see that the dynamic failure ratios of DSP are the highest, that of DWCS and EDF are nearly similar and that of TMS are the lowest. With the utilization factor of a set increasing, the performance of TMS will reduce. Though the performance of TMS will decrease rapidly when the utilization factor of a set is above 1.9, it still surpasses the other algorithms.

Hence, TMS outperforms DWCS, BMS, EDF, and DSP.

Table 7. Breakdown utilization factor of sets scheduled by TMS and BMS (experiment 1)

	Average utilization factors	Average breakdown utilizations	Average Increased value
TMS	0.89152	1.5738	0.68228
BMS	0.89152	1.1554	0.26388

Table 8. Breakdown utilization factor of sets scheduled by TMS and BMS (experiment 2)

	Average utilization factors	Average breakdown utilizations	Average Increased value
TMS	0.83985	1.2742	0.43435
BMS	0.83985	0.9565	0.11665

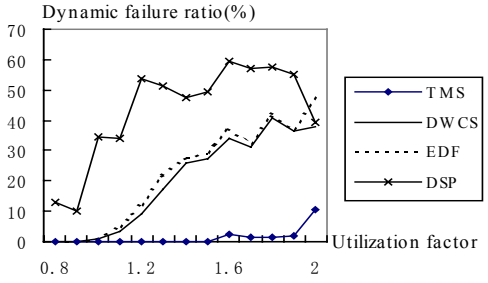


Fig. 3. Comparison of the dynamic failure ratio (experiment 1)

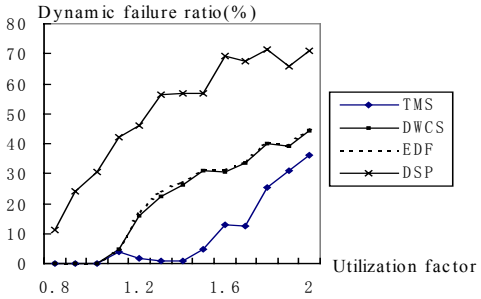


Fig. 4. Comparison of the dynamic failure ratio (experiment 2)

5. CONCLUSION

In this paper, a new online scheduling algorithm, called TMS, is proposed for sets of periodic tasks with (m,k) deadlines in overloaded real-time systems. It combines the states and deadlines of tasks seamlessly in order to overcome the shortcomings of the existing algorithms. The priorities of requests in the critical states are the highest. Priority of a request in the quasi-critical state will be heightened, thereby avoiding it going to the critical state. The priorities of tasks in the normal states are the lowest. Finally, experiments were carried out to show that the performance of TMS is much better than that of DWCS, EDF, DSP and BMS. TMS are suitable for scheduling both tasks with (m,k) deadlines and soft real-time tasks.

6. REFERENCES

[1] M. Hamdaoui, P. Ramanathan, “A Dynamic Priority Assignment Technique for Streams with (m,k)-firm Deadlines”, IEEE Transactions on Computer, Dec. 1995, 44(12): 1443-1451.

[2] G. Koren, D. Shasha, “Skip-Over: Algorithms and Complexity for Overloaded Systems that Allow Skips”, Proceedings of the 16th IEEE Real-Time Systems Symposium, Pisa, Italy, Dec. 1995, 110-117.

[3] G. Bernat, A. Burns, “Combing (m, n)-hard deadlines and dual priority scheduling”, Proceedings of the 18th IEEE Real-Time Systems Symposium, IEEE Computer Society, San Francisco, 1997.

[4] G. B.t Nicolau, “Specification and analysis of Weakly Hard Real-time Systems”, PHD thesis, Univerisitat de les llles Barlears Department de Ciències Matemàytiques I informàtica, Spain, 1998.

[5] G. B. Nicolau, “Weakly hard real-time systems”, IEEE transactions on Computers, April 2001, 50(4): 308~321.

[6] R. West, C. Poellabauer, “Analysis of a Window-Constrained Scheduler for Real-Time and Best-Effort Packet Streams”, Proceedings of the 21st IEEE Real-Time Systems Symposium (RTSS 2000), Orlando, FL, November 2000.

[7] K. Jeffay, D. Stone, “Accounting for Interrupt Handling Costs in Dynamic Priority Task Systems”, Proceedings of the 14th IEEE Symposium on Real-Time Systems, Dec. 1993, 212-221.

[8] J. Lehoczky, L.SHA, et al, “The rate monotonic scheduling algorithm: exact characterization and average case behavior”. Proceedings of the IEEE Real-Time Systems Symposium (RTSS1989), Santa

Monica, California, USA, 1989, 166-171.

[9] P Ramanathan, "Overload Management in Real-Time Control Applications Using (m, k)-Firm Guarantee", IEEE Transactions on Parallel and Distributed Systems, 1999, 10(6): 549-559.

[10] G. Quan, X.B. Hu, "Enhanced Fixed-Priority Scheduling with (m, k)-firm Guarantee", 21st IEEE Real-Time Systems Symposium, Florida, USA, December 2000, 79-88.

[11] J. Goossens, R. Devillers, "Feasibility Intervals for the Deadline Driven Scheduler with Arbitrary Deadlines", Proceedings of the sixth international conference on real-time computing systems and applications (RTCSA'99), Hong Kong, China, December 1999, 54-61.

[12] C. L Liu, J. Layland, "Scheduling Algorithm for Multiprogramming in a Hard Real-Time Environment", Journal of the ACM, January 1973, Vol.20: 46-61.

[13] L. George, N. Rivierre, M. Spuri, "Preemptive and non-preemptive real-time uniprocessor scheduling", Technic report 2966. INRIA Rocquencourt, 1996.

[14] Y. Wang, M. Saksena, "Scheduling Fixed-Priority Tasks with preemption threshold", Proceedings IEEE International Conference on Real-Time Computing Systems and Applications, Hong Kong, December 1999.

Solving Optimization of System Reliability by Ant Colony Algorithm

Shang Gao1,2
1 School of electronics and information, Jiangsu University of Science and Technology, Zhenjiang 212003,China;
2 Provincial Key Laboratory of Computer Information Processing Technology, Soochow University ,Suzhou 215006,China
Email: gao_shang@hotmail.com

Lingfang Sun
School of Economic and management, Jiangsu University of Science and Technology, Zhenjiang 212003,China
Email: slf0308@163.com

Xinzi JIANG
School of electronics and information, Jiangsu University of Science and Technology, Zhenjiang 212003,China

Kezong TANG
School of electronics and information, Jiangsu University of Science and Technology, Zhenjiang 212003,China

ABSTRACT

A redundancy optimization model is given in this paper. Many optimization methods to solve optimum model and their advantages and shortages are analyzed. An ant colony algorithm is put forward comparing with heuristic method, and its effectiveness is illustrated through result.

Keywords: Ant colony algorithm, Reliability, Optimization, Heuristic method.

1. INTRODUCTION

This paper describes the use of an Ant colony optimization (ACO) algorithm to solve the redundancy allocation problem for a series-parallel system. The reliable performance of a system for a predefined mission time under various conditions is very important in many industrial and military applications. In the formulation of a series-parallel system problem, for each subsystem, multiple component choices (assuming an unlimited supply of each component) are used in parallel. For those systems, with known cost, reliability, and weight, system design and component selection becomes a combinatorial optimization problem. The problem is then to select the optimal combination of parts and redundancy levels to meet cost and weight constraints collectively while maximizing system reliability.

2. REDUNDANCY OPTIMIZATION MODEL

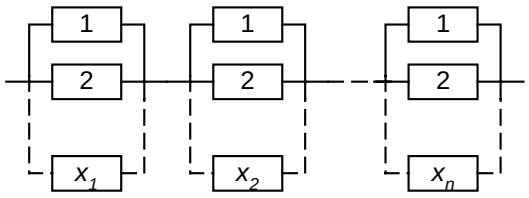


Fig. 1. Series-parallel system configuration

We assume a mixed series-parallel system has n stages. Our aim is to optimal the cost of system subject to constraint on system reliability. The schematic is in Fig. 1. Find the x_i, $i = 1,2,\cdots,n$

minimize: $C_s = \sum_{i=1}^{n} c_i x_i$

subject to: $\prod_{i=1}^{n} R_i(x_i) \geq R_0$ (1)

where C_s =cost of system. R_s =reliability of system. c_i =cost of component i. x_i =number of redundancies of component i, $x_i \geq 1$. $R_i(x_i)$ =reliability of stage i, $R_i(x_i) = 1-(1-p_i)^{x_i}$. R_0 =lower bound on R_s. p_i =reliability of component i.

The problem is NP-hard, only some special cases can be solved efficiently. Thus, many approaches have been proposed to find optimal design in reasonable time. Tillman, Hwang, and Kuo [1] provide a thorough review related to optimal system reliability with redundancy. They divided optimal system reliability models with redundancy into series, parallel, series-parallel, parallel-series, standby, and complex classes. They also categorized optimization methods into integer programming, dynamic programming, linear programming, geometric programming, generalized Lagrangian functions, and heuristic approaches. The authors concluded that many algorithms have been proposed but only a few have been demonstrated to be effective when applied to large-scale nonlinear programming problems. Also, none has proven to be generally superior. Fyffe, Hines, and Lee[2] provide a dynamic programming algorithm for solving the system reliability allocation problem. As the number of constraints in a given reliability problem increases, the computation required for solving the problem increases exponentially. In order to overcome these computational difficulties, the authors introduce the Lagrange multiplier to reduce the dimensionality of the problem. Nakagawa and Miyazaki[3] show a more efficient algorithm compared to dynamic programming using the Lagrange multiplier. In their algorithm, the authors use surrogate constraints obtained by combining multiple constraints into one constraint. Misra and Sharma[4] present a simple and efficient technique for solving integer

programming problems such as the system reliability design problem.Hwang, Tillman and Kuo[5]use the generalized Lagrangian function method and the generalized reduced gradient method to solve nonlinear optimization problems for reliability of a complex system. The same authors also present a mixed integer programming approach to solve the reliability problem. They maximize the system reliability as a function of component reliability level and the number of components at each stage. Using a genetic algorithm (GA) approach, Coit and Smith[6,7,8] provide a competitive and robust algorithm to solve the system reliability problem. The authors use a penalty guided algorithm which searches over feasible and infeasible regions to identify a final, feasible optimal, or near optimal, solution. For a fixed design configuration and known incremental decreases in component failure rates and their associated costs, Painton and Campbell[9] also use a GA to find a maximum reliability solution to satisfy specific cost constraints.

3. METHODOLOGY

3.1 ACO algorithm: Literature Review

Inspired by the collective behavior of a real ant colony, Marco Dorigo first introduced the Ant System (AS) in his Ph.D. thesis[10], and the study was further continued by Dorrigo, Maniezzo, and Colorni[11]. The characteristics of an artificial ant colony include positive feedback, distributed computation, and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search process. In order to demonstrate the ACO approach, the authors apply this approach to the classical TSP, asymmetric TSP, QAP, and job-shop scheduling. AS shows very good results in each applied area. More recently Dorigo and Gambardella[12,13] have been working on extended versions of the ACO paradigm. Ant Colony System (ACS) is one of the extensions and has been applied to the symmetric and asymmetric TSP with excellent results. The Ant System has also been applied with success to other combinatorial optimization problems such as the vehicle routing problem, telecommunications networks, scheduling, graph coloring, partitioning, etc.

3.2 Ant Colony Algorithm for Optimization of System Reliability

Firstly, we transform (1)(constrained problem) into a single unconstrained problem.

$$\min \sum_{i=1}^{n} c_i x_i + M\left\{\min\left\{0,\left[\prod_{i=1}^{n}(1-R_i)^{x_i} - R_0\right]\right\}\right\}^2 \qquad (2)$$

where $M > 0$ is a large number.

As the overall restrictions on system cost, volume and weight, the maximum number of redundant component is $x_{\max}$. The reliability configuration turns into reliability network. Each stage has $x_{\max}$ notes. Each note represents the number of redundant component. If we connect these notes, we will get a solution. Fig. 2. represents a solution $(3,2,1,\cdots,1)$.

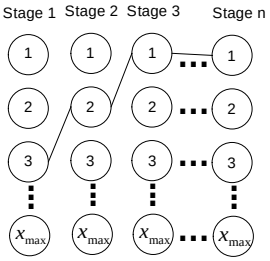


Fig. 2. Reliability network

Each stage has an ant, and each ant can transfer from the first note to the $x_{\max}$ th note only in self stage. The transition probability for ant to go from the i th note to the j th note is:

$$p_{ij} = \tau_{ij} / \sum_{i=1}^{x_{\max}} \tau_{ij} \qquad (3)$$

Here τ_{ij} represents the desire of setting stage j at the i th note.Thus, the trail intensities are updated according to the formula

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} + Q \qquad (4)$$

where ρ with $0 < \rho < 1$ is the persistence of the trail, thus $1-\rho$ represents the evaporation. Q is a constant related to the magnitude of the trail laid by ants.

η_{ij} represents the margin of objective function. If $\eta_{ij} > 0$, it mean that objective function reduce. Thus The ant goes from the i th note to the j th note .Otherwise $\eta_{ij} \le 0$, the ant remain previous position.

Formally the ant colony algorithm is:

Step 1. Set $nc = 0$, and assign the same value to τ_{ij}.

Step 2. Put n ants into first note of every stage. That represents $(x_1, x_2, \cdots, x_n) = (1,1,\cdots,1)$.

Step 3. Choose the next note to move with transition probability p_{ij}. If $\eta_{ij} \le 0$, the ant remain previous position.

Step 4. Update trail values and the transition probability, and set $nc = nc + 1$.

Step 5.If ($nc < nc_{\max}$) then goto step 3, else print component selection and stop.

4. COMPARISON OF ACO WITH HEURISTIC METHOD

We use a heuristic method to solve this problem.the algorithm is as follows：

Step1.Let $(x_1, x_2, \cdots, x_n) = (1,1,\cdots,1)$.

Step2.Set $D_i(x_i) = [\ln R_i(x_1 + 1) - \ln R_i(x_1)] / c_i$.

Find integer i_0 satisfying $D_{i_0}(x_{i_0}) = \max_i [D_i(x_i)]$.

Replace x_{i_0} by $x_{i_0} + 1$,go to step 3.

Step3.If $R_s = \prod_{i=1}^{n}[1-(1-p_i)^{x_i}] \geq R_0$,return. Otherwise go back step 2.

In order to compare ACO to heuristic method, an example is tested. $p_1 = 0.96$, $p_2 = 0.93$, $p_3 = 0.85$, $p_4 = 0.80$, $p_5 = 0.75$, $c_1 = 3$, $c_2 = 12$, $c_3 = 8$, $c_4 = 5$, $c_5 = 10$, $R_0 = 0.9$. Table 1. shows the solution of heuristic method.

Table 1. The solution of Heuristic method

step	x_1	x_2	x_3	x_4	x_5	C_s	R_s
1	1	1	1	1	1	38	0.4553
2	1	1	1	2	1	43	0.5463
3	1	1	1	2	2	53	0.6829
4	1	1	2	2	2	61	0.7854
5	2	1	2	3	2	64	0.8169
6	2	2	2	3	2	69	0.8441
7	2	2	2	3	2	81	0.9032

We set ACO parameters in the following way: $x_{\max} = 4$, $[\tau_{ij}]_{4\times5} = [10]_{4\times5}$, $M = 10^6$, $\rho = 0.9$, $Q = 10$. The best solution is $(2,2,2,3,2)$. The minimum cost of system is 81. By the way, result is may be different after each run, but it is a satisfaction solution. Fig. 3. shows the cost of system during iterations Using ACO. In heuristic method, heuristic gene is determined by c_i and p_i , and the convergence rate lies on Heuristic gene. In ACO, c_i and p_i are joined to penalty function, we needn't consider them in computation. If n is small, heuristic method is better than ACO, otherwise ACO is better.

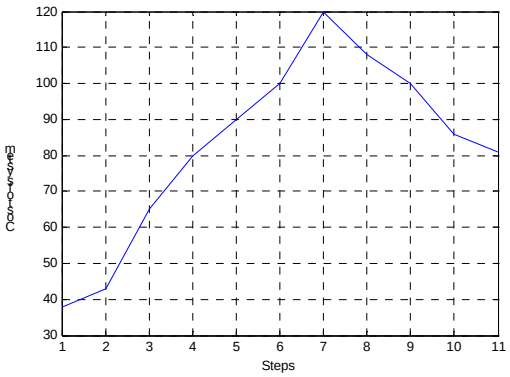


Fig. 3. Computing process of ACO

5. CONCLUSIONS

The ACO is a promising method for solving combinatorial optimization problems. The efficiency of this algorithm will depend on the selection of parameters; trail update method, and other supplemental devices. In this paper, we show how to apply the ACO to the redundancy allocation problem.

REFERENCES

[1] A.F. Tillman, C.L. Hwang, et al, "Optimization Techniques for System Reliability with Redundancy - A Review", IEEE Transactions on Reliability, 1977,R-26（3）: 148-155.

[2] E. D. Fyffe, W.W. Hines, et al, "System Reliability Allocation And a Computational Algorithm", IEEE Transactions on Reliability, 1968, R-17（2）: 64-69.

[3] Y.J. Nakagawa, S. Miyazaki, "Surrogate Constraints Algorithm for Reliability Optimization Problems with Two Constraints", IEEE Transactions on Reliability, 1981, R-30(2) :175-180.

[4] K.B. Misra, U. Sharma, "An Efficient Algorithm to Solve Integer-Programming Problems Arising in System-Reliability Design", IEEE Transactions on Reliability, 1991, 40(1): 81-91.

[5] A.F. Tillman, C.L. Hwang, et al, "Determining Component Reliability and Redundancy for Optimum System Reliability", IEEE Transactions on Reliability, 1977,R-26（3）: 162-165.

[6] W. David. Coit, E. Alice. Smith, "Reliability Optimization of Series-Parallel Systems Using a Genetic Algorithm", IEEE Transactions on Reliability, vol. 45, no. 2, 1996, 254-260.

[7] W. David. Coit, E. Alice. Smith, "Penalty Guided Genetic Search for Reliability Design Optimization", Computers and Industrial Engineering, 1996, 30（4）: 95-904.

[8] W. David. Coit, E. Alice. Smith, et al, "Adaptive Penalty Methods for Genetic Optimization of Constrained Combinatorial Problems", INFORMS Journal on Computing, 1996，8（2）: 173-182.

[9] L. Painton, J. Campbell, "Genetic Algorithms in Optimization of System Reliability, IEEE Transactions on Reliability", 1995, 44（2）:172-178.

[10] M. Dorigo, V. Maniezzo, et al, "Positive Feedback as a Search Strategy", Technical Report No. 91-016, Politecnico di Milano, Italy. 1991

[11] M. Dorigo, V. Maniezzo, et al, "The Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996，26（1）:29-41.

[12] M. Dorigo, L. M. Gambardella, "Ant Colonies for the Travelling Salesman Problem", BioSystems, vol. 43, 1997， 73-81.

[13] M. Dorigo, L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem", IEEE Transactions on Evolutionary Computation, 1997, 1（1）: 53-66.

The Influences of the Two Bound Treatments on the Performance of IIR Filter Design Based on Particle Swarm Optimization

Wei Fang, Wenbo Xu
Institute of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, 214122, China
Email: {wxfangwei, sunjun_wx, xwb_sytu}@hotmail.com

ABSTRACT

Digital IIR filter is an important kind of filter and has been an active area of research for many years. Particle Swarm Optimization (PSO) is a novel random searching optimization algorithm and can reach the global optimization solution. We apply PSO to design digital IIR filter and propose a new bound treatment approach. Compared with the origin bound treatment approach, our proposed method in design digital IIR filter can get better performance. Experiment results show that the modified algorithm outperforms the origin PSO.

Keywords: Particle Swarm Optimization, IIR filter, filter design

1. INTRODUCTION

Digital IIR filter is an important kind of filter. They are broadly used in the communication, voice signal and image signal processing, HDTV, biomedicine and seismic survey etc. There are two commonly methods in design digital IIR filters. The most commonly method is design an analog IIR filter first and then transform the analog filter to the digital IIR filter. The second method employs algorithm design approaches to design filters, such as least mean square error method, least p error method and etc. The essence of this method is to obtain the optimal performance filters under certain restriction. [1]

Recently, researchers have applied global optimization approaches to digital IIR filter design, such as Genetic Algorithm (GA)[2], Particle Swarm Optimization (PSO)[3], etc. But as we know, it is difficult to use GA for its complex coding and algorithm structure, and slow computation. PSO was originally proposed by J.Kennedy and R.Eberhart [4][5], has become an important global optimization technique. But convergence rate of the algorithm may become slowly if unreasonable method used in treating the bound when it is applied in design IIR filters.

The paper is organized as follows. In section 2, the problem that will be discussed is put. In section 3, PSO and two bound treatments in PSO are introduced. In section 4, the application of PSO to digital IIR filter design is presented and the experiment is in section 5. A conclusion is found in section 6.

2. OPTIMIZATION PRINCIPLE IN DESINGNING IIR FILTERS[6]

As we know, the essence of design IIR filter is to determine the coefficients $\{a_k\}$ and $\{b_k\}$ of its system function, that is

$$H(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{1 - \sum_{k=1}^{N} a_k z^{-k}} \quad (1)$$

As first-order and second-order digital filters are simple and easy to design and insensitive to limited word length, it is an available method to use them design high order filters. Then system functions can be described as:

$$H(z) = A\prod_{k=1}^{K} H_k(z), \quad (2)$$

where

$$H_k(z) = \frac{1 + a_k z^{-1} + b_k z^{-2}}{1 + c_k z^{-1} + d_k z^{-2}}. \quad (3)$$

in the eq.3, there are four coefficients a_k, b_k, c_k, d_k to be determined and there are 4k+1 coefficients in the system function to be obtained, they are A, a_k, b_k, c_k, d_k, k=1,…K.

From the system function (1), the frequency response of an IIR system can be written as:

$$H(e^{jw_i}) = H(z) \mid z = e^{jw_i}, \quad (4)$$

where w_i is a set of separate points and i= 1,2,…M. On these separate points, the frequency response error between the actual filter and the desired filter is:

$$Ei = |H(z_i)| - |H_d(z_i)|. \quad (5)$$

Then the total error Q at all the frequencies is:

$$Q = \sum_{i=1}^{M} E_i^2 = \sum_{i=1}^{M} [|H(z_i)| - |H_d(z_i)|]^2. \quad (6)$$

The design criterion is to find a set of coefficients $\{A, a_1, b_1, c_1, d_1, \cdots, a_k, b_k, c_k, d_k\}$ in order to minimize the value of Q. In other words, make the magnitude of *H(z)* approaches the magnitude of $H_d(z_i)$ as far as possible, expressed by,

Minimize Q; (7)

Assume the vector

$$X = (a_1, b_1, c_1, d_1, \cdots, a_k, b_k, c_k, d_k)^T, \quad (8)$$

then the dimension of *X* is 4k. The transfer function $H(z)$ can be rewrite as follows:

$$H(z) = A\prod_{k=1}^{k} H_k(z) = A\prod_{k=1}^{k} \frac{1 + a^k z^{-1} + b^k z^{-2}}{1 + c^k z^{-1} + d^k z^{-2}} = AH(z, X) \quad (9)$$

and using this definition we can write equation (6) as

$$Q(A, X) = \sum_{i=1}^{M} [|AH(z_i, X)| - |H_d(z_i)|]^2 \quad (10)$$

In order to get the optimal value A^0 of A, we let

$$\partial Q(A, X) / \partial A = 0 \quad (11)$$

and substitute the formula above into this it, we obtain

$$|A| = \frac{\sum_{i=1}^{M} |H(z, X)| \bullet |H_d(z_i)|}{\sum_{i=1}^{M} |H(z_i, X)|^2} = A^0 . \quad (12)$$

As we just consider the magnitude error, the sign of A doesn't consider. Then the formula can be simplified as

$$Q_1(X) = Q(A^0, X). \quad (13)$$

The goal is to solve the formula above and get the minimum and the minimum point X^0.

3. PARTICLE SWARM OPTIMIZATION

3.1 Introduction

PSO, originally proposed by J.Kennedy and R.Eberhart [4][5], is a new global search technique. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory. Like genetic algorithm (GA), PSO is a population-based random search technique and outperforms GA in many practical applications, particularly in nonlinear optimization problems. PSO has become an important optimization implement since it has less parameter, simplicity in software programming and the fast convergence rate.

In the standard PSO model, each individual is treated as a volume-less particle in the D-dimensional space, with the position and velocity of *i*th particle represented as $X_i = (X_{i1}, X_{i2}, \cdots, X_{iD})$ and $V_i = (V_{i1}, V_{i2}, \cdots, V_{iD})$. The particles moves according to the following equation:

$$V_{id} = w * V_{id} + c_1 * rand(\cdot) * (P_{id} - X_{id}) + c_2 * Rand(\cdot) * (P_g - X_{id}) \quad (14)$$

$$X_{id} = X_{id} + V_{id} \quad (15)$$

where c_1 and c_2 are positive constant and rand() and Rand() are two random functions in the range of [0,1]. Vector $P_i = (P_{i1}, P_{i2}, \cdots, P_{iD})$ is the best previous position (the position giving the best fitness value) of particle *i* called pbest, and vector $P_g = (P_{g1}, P_{g2}, \cdots, P_{gD})$ is the position of the best particle among all the particles in the population and called gbest. Parameter *w* is the inertia weight introduced to accelerate the convergence speed of the PSO. Usually parameter w decreases linearly according to the following formula [7]

$$w = w_{\max} - iter * (w_{\max} - w_{\min}) / iteration \quad (16)$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum value of w respectively, which always take values 0.9 and 0.4 , iteration is the number of evolutions, iter is currently iteration of the evolution. From equation (16), when the evolution begins, particles have a large search space. As the evolution goes on, parameter *w* gets small, and local convergence capacity enhanced.

3.2 Two bound treatment

In PSO, the position of particles may exceed the problem space in each iteration. When this kind of situation occurs, it's necessary to treat. Usually, if the current position is bigger than the upper limits of problem space, the upper limit is set as the current position, that is

$$x = x_{\max}, \quad (17)$$

If the current position is smaller than the lower limit of problem space, the lower limit is set as the current position, that is,

$$x = x_{\min} \quad (18)$$

where x is the current position of the particles.

This kind of bound treatment may result in insufficiency optimization of filter's coefficients; in other words, many coefficients only can obtain the upper limit or lower limit and the performance of the designed filter is poor.

Then in the paper, a novel method is proposed to solve this problem. Consider the current position of particles, whether it exceeds the upper limit or the lower limit, we the current position by its reciprocal, that is,

$$x = 1/x. \quad (19)$$

Based on this method, the particles exceed the limits, they will be restricted within the problem space and will accelerate the convergence speed. Simultaneity, the modified algorithm will not easily get the local optimization.

4. APPLYING THE ALGORITHM IN THE DESIGN OF IIR FILTERS

4.1 Parameter Coding

In PSO, real number can be coding as the parameters in the algorithm directly and needn't decode. Our problem is to obtain the parameters in the equation (13). Then the position of particles will be coded in the algorithm as the coefficients of the desired filter. Also in PSO, the position of each particle has a corresponding velocity.

4.2 Parameter bounds

In order to keep the stability of the filter when designing the filter, it is necessary to enable each second order section of filter lie in the unit circle of *z*-plane, that is, set the poles z_k of the denominator in Eq.(3) satisfy $|z_k| < 1$. Then we can get that

$$-2 < c_k < 2, -1 < d_k < 1, K = 1, 2, \cdots, N .$$

Even in the range, there exists the poles that make the filter instability, we can replace them by their reciprocal and unchanged the magnitude response a stability filter can be designed. In addition, if the least phase shift characteristics is needed, we can deal with a_k, b_k just like c_k, d_k.

4.3 Fitness function

According to the design essence, Eq.(13) is selected as the fitness function, that is,

$$Fitness = Q(A^0, X).$$

From the formula, it can be seen that the smaller the fitness function value is, the better performance the filter has.

4.4 Design procedure

Step1: Given the desired target of the filter
Step2: set the parameters of the algorithm; include the number of the particles, dimension of the parameters
Step3: initial the position and velocity of the particles
Step4: calculate the fitness function value of each particle

Step5: update the current optimal solution of particles and the global optimal solution of all the particles
Step6: recalculate the fitness function value of each particle
Step7: judges whether update the current optimal solution and global optimal solution
Step8: loop from step 4 to step 7 until a certain condition satisfied
Step9: output the optimal solution, that is the coefficient of the desired filter

5. EXPERIMENTS

We set down the modified PSO as PSO_M. The results are obtained after 50 independent experiments. Genetic Algorithm (GA) is also employed to design the filter for the comparison with the performance of modified PSO, which takes two different bound treatments. The settings of experiments are as follows:

GA: tournament selection, uniform crossover，uniform mutation，pop size =50; Iteration=500; Crossover Rate=0.8; Mutation Rate=0.05.

PSO: pop size=50; maximum velocity=4; w_{max}=0.9, w_{min}=0.4.

We will design an IIR lowpass digital filter; the frequency response of the filter is as follows:

$$H_d(e^{jw}) = \begin{cases} 1 & 0 \le w \le 0.4\pi \\ 0.5 & w = 0.45\pi \\ 0 & 0.5\pi \le w \le \pi \end{cases}$$

we use three second-order sections to design and the number of frequency sampling points M=23.

Fig1, 2 show the comparison of averaged convergence rate and magnitude response of three methods. Table1 shows the coefficients and the optimal value obtained by three methods.

From fig1, it can be seen that the optimal value obtained by the modified PSO is much smaller than the other two. In fig2, we can conclude that the performance of the filter designed by the modified PSO outperforms the other two.

6. CONCLUSIONS

Based on the PSO, we proposed the modified algorithm aimed at the treatment of the bound of particles position. And we employed the modified algorithm in the design of digital IIR filters. The modified algorithm can get the more optimal value than PSO and GA. Compared with the results of using GA and PSO, it was concluded that the modified algorithm could be efficiently used for design digital IIR filter design.

7. REFERENCES

[1] P.Q Chen, Digital signal processing (Second Edition), pp.315-322, Aug, 2001.
[2] J.H.Li, F.L.Yin, “Genetic Algorithms for Designing IIR Digital Filters”, Journal of china institute of communications, Vol. 17, No3. May 1996.

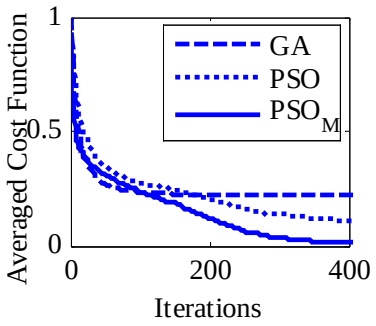


Fig 1. Convergence behaviors of three methods

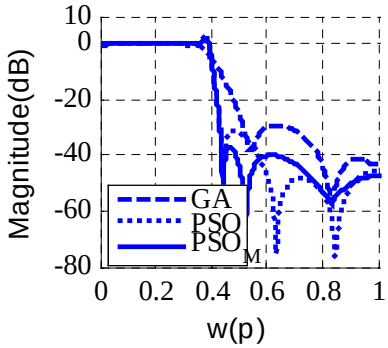


Fig 2. Magnitude response of three methods

TABLE I
Coefficients of the designed filer by three methods

	GA	PSO	PSO_M
A_0	0.4698	0.09237	0.04512
a1	0.2612	-0.3542	1.5362
b1	1.0945	0.9813	0.7973
c1	-1.0007	-0.8060	-0.6462
d1	1.5009	0.5116	0.5980
b1	-0.1324	1.7545	-0.3570
b2	-1.5389	0.9811	0.9947
b3	1.5634	-0.6868	-0.8314
b4	-1.665	0.9351	0.2697
c1	1.7744	0.8152	0.1432
c2	1.0476	0.9900	0.9852
c3	-1.999	-1.7895	-0.6471
c4	-1.1766	-0.9781	0.9183
MSE	0.2205	0.0934	0.0047
Run Time(s)	45.55	16.99	15.04

[3] Z.R.Hou, Z.S.Lu, “Particle Swarm Optimization Algorithm for IIR Digital Filters Design”, Journal of circuts and systems. Vol.8, No.4, August 2003.
[4] J. Kennedy, R. Eberhart, “Particle Swarm Optimization,” Proc. IEEE int. Conf. On Neural Network, J, 1995, pp. 1942-1948
[5] Shi Y, R.C.Eberhart, “A Modified Swarm optimizer,” .Proc. IEEE int Conf of Evolutionary Computation.1998,
[6] Y.H. Ni, C.G.Wang, Digital Signal Processing—Theory and Realization, 1998
[7] Y. Shi, R.C. Eberhart, “A Modified Swarm optimizer,”.Proc. IEEE int Conf of Evolutionary Computation.1998, pp.69-73.

Advanced Strategy of Genetic Algorithm Adaptive Immunity Technology

Xinchi Yan
Department of Civil Engineering, Southern Yangtze University
Wuxi 214122, China
Email: ellayan@citiz.net

ABSTRACT

Genetic Algorithm often gets convergence prematurely, and classical adaptive technology still has much defect in real complex structure. Therefore, advanced adaptive technology with distant kinsman pointer was presented to break the phenomenon that some single individual was full of the population, and immunity operator was adopted to prevent individual degeneration in crossover or mutation. The conformation and capability of "distant kinsman pointer" and "immunity operator" were circumstantiated. These ensure the algorithm can get global optimal solution quickly; this method is called Advanced Adaptive Immunity Genetic Algorithm. The numeral example shows that this advanced strategy can make GA has better convergence efficiency in Reliability-Based Structural Optimization.

Keywords: Genetic Algorithm, Reliability-Based Structural Optimization; Adaptive Technology; Immunity Operator; Distant Kinsman Pointer.

1. INTRODUCTION

In Standard Genetic Algorithm (SGA)[1,2], the selection, crossover and mutation are always work in stochastic condition[3], unavoidable, some individuals will be degenerate in the circulate course, that will debase GA's capability, so, add immunity operator to advance the algorithm's capability, and to restrain the degeneration of individuals. Crossover probability p_c and mutation probability p_m influence the algorithm's capability, Literature[4] presents the dynamic Adaptive strategy, but some calculations show that it can't be satisfied.

This paper adopts distant kinsman pointer adaptive strategy and immunity operator to ensure the algorithm's global optimal solution convergence, which called Advanced Adaptive Immunity Genetic Algorithm (AAIGA).

2. ADAPTIVE STRATEGY WITH DISTANT KINSMAN POINTER

Typical adaptive strategy can make p_c and p_m change according to individual's fitness. When the fitness of the individuals become accordant, it will make p_c and p_m increase, when the fitness are disperse, it will make p_c and p_m decrease. At the same time, for the individuals whose fitness is more than the population average fitness, it will give a lower p_c and p_m to protect them, and for the individuals whose fitness is less than the population average fitness, it will give bigger p_c and p_m.

$$p_c = \begin{cases} \dfrac{k_1(f_{\max} - f^*)}{f_{\max} - \overline{f}}, & f^* \geq \overline{f} \\ k_2, & f^* < \overline{f} \end{cases} \quad (1)$$

$$p_m = \begin{cases} \dfrac{k_3(f_{\max} - f)}{f_{\max} - \overline{f}}, & f \geq \overline{f} \\ k_4, & f < \overline{f} \end{cases} \quad (2)$$

where $f_{\max}$ = population maximal fitness; $\overline{f}$ =population average fitness; f^* =one of biggish crossover individuals fitness; f =mutation individual fitness; $k_1, k_2, k_3, k_4 \in (0,1)$.

Equations(1) and (2) show when population fitness go to accordant, AGA make p_c and p_m increase. But this kind of accord is possibly means some strong individuals full of the population, typical adaptive strategy can't control this question. So, whether the algorithm can jump from partial optimal depend on these extra strong individual reproduction in the population must be restricted, thus depend on distant kinsman pointer work in early time of selection and crossover.

Distant kinsman pointer works as follows:

Given that calculation is doing the *t*th generation, the population is $P(t) = \{x_i\}$. If one individual were selected two times in the $t+1$ th generation, then remember its' location number, and set a distant kinsman pointer.

First, if the individual, who has distant kinsman pointer, was selected to be crossover, it must couple with different one. Then, another individual go to the work of immunity operator. The other individuals' p_c and p_m of the population still calculate as Equations(1) and (2).

It is clear that distant kinsman pointer can prevent individual full of the population, improve the algorithm's capability of creating new mode, and it can be control very easy. Distant kinsman pointer keeps population diversity and ensures the algorithm's global convergence.

3. IMMUNITY OPERATOR

In order to prevent individual degenerate in basic operators and adaptive strategy work, immunity operator was adopted, and this operator make up of inoculating vaccine and immunity selection.

1) Inoculate vaccine

Vaccine is some basic information of the question, in here; it can be some gene of the best individual. Given that calculation is doing the *t*th generation, population is $P(t) = \{x_i\}$, choose random $m(m \leq n)$ individuals to inoculate vaccine. m is calculated with the following normalized equations:

$$m = a \times n \quad (3)$$

where $a = 0.05 \sim 0.1$ is inoculate probability, n is Population size.

Extract and inoculate process are shown in Fig. 1 and Fig. 2.

2）Immunity selection

Inspect the individuals that have been inoculated vaccine, if their fitness were less than their father's, they would be replaced by their father individuals, if not, that means inoculated success, new partly optimal individual appear, and select it in new generation.

It is clear that immunity operator can improve partly search capability, prevent individual degenerate, and ensure distant kinsman pointer adaptive strategy work powerfully.

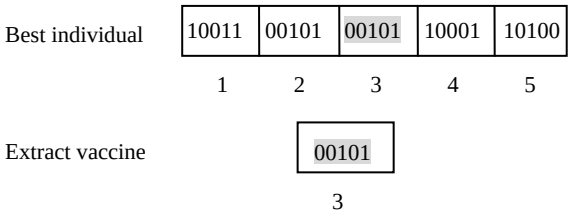


Fig. 1 Extract vaccine

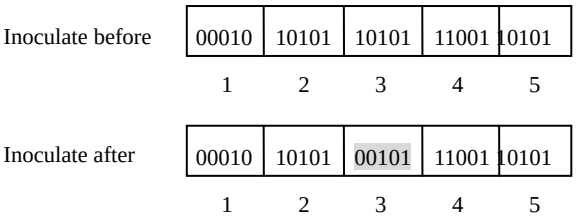


Fig. 2 Inoculate vaccine

Then, the advanced adaptive strategy has finished; an Advanced Adaptive Immunity Genetic Algorithm (AAIGA) has been made.

4. NUMERAL EXAMPLE

A classical 25-bar truss RBSO problem is present as an illustrative example by using AAIGA.

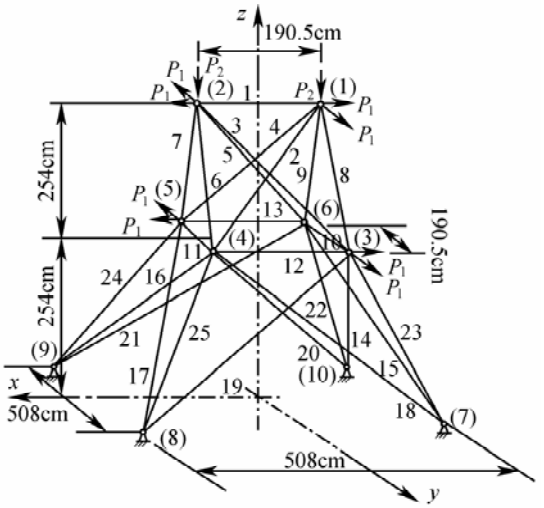


Fig. 3. 25-bar truss structure

The details of the truss geometry and loading are shown in Fig. 3 $\overline{C}_Y = 27.6kN/cm^2$, $Cov = 0.05$. $E = 6895kN/cm^2$, $\rho = 2.7\times10^{-3}kg/cm^3$. loads $\overline{P}_1 = 88.9kN$, $\overline{P}_2 = 22.6kN$, $Cov = 0.2$. The target system reliability index is $\beta_s^a = 3.5$. bars' cross-sectional areas are A_i（0.5~15）cm^2.

Member grouping details are I（A_1），II（$A_2 = A_5$），III（$A_3 = A_4$），IV（$A_6 = A_9$），V（$A_7 = A_8$），VI（$A_{10} = A_{11}$），VII（$A_{12} = A_{13}$），VIII（$A_{14} = A_{17}$），IX（$A_{15} = A_{16}$），X（$A_{18} = A_{21}$），XI（$A_{19} = A_{20}$），XII（$A_{22} = A_{25}$），XIII（$A_{23} = A_{24}$）。

The RBSO question can illustrate as follows:

$$\begin{cases} \min W = \sum_{i=1}^{25} A_i L_i \rho_i \\ s.t. \quad 0.5 \le A \le 15 \\ \beta_s \ge 3.5 \end{cases}$$

Exterior penalty method in literature[5] was adopted to deal with the constraints, where $C = 1.25\times$ structure system weights when A_i equal maximal value, then $C = 416.84$. Fitness function is

$$F = 416.84 - W(1 + k_1 g)$$

where $g = \begin{cases} 1 - \dfrac{\beta_s}{3.5}, & \beta_s < 3.5 \\ 0, & \beta_s \ge 3.5 \end{cases}$

The termination criterion is max generation not more than 110. The control parameters are shown in Table 1., and result is given in Table 2.

Table 1. Control parameters of AAIGA

Control parameters	
Population size n	30
Bit string length l_s	10
Individual bit string length l	130（13×10）

Table 2. Optimization result of 25-bar truss

Element group	Cross-sectional areas（cm^2）	Element group	Cross-sectional areas（cm^2）
I	5.618	II	5.240
III	4.649	IV	4.944
V	9.732	VI	5.236
VII	4.156	VIII	4.156
IX	4.156	X	4.156
XI	4.156	XII	4.643
XIII	4.640		
System weight	111.158(kg)	β_s	3.500

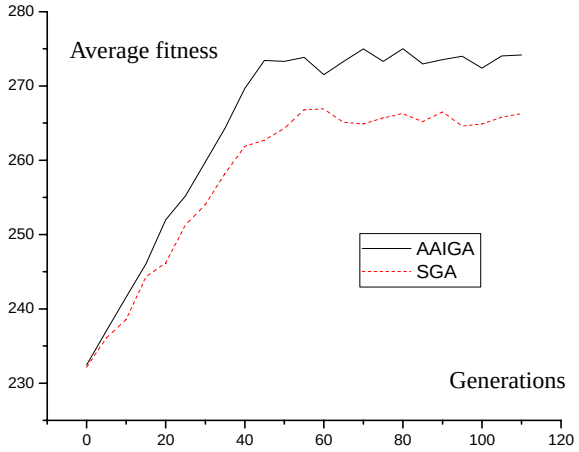


Fig. 4. Average fitness of every generation

From Fig. 4., it is observed that AIGA has preferable capability and is effective in random structure optimization.

5. CONCLUSIONS

Under the same conditions, 25-bar truss has been calculated by use SGA and AAIGA. From the average fitness of every generation, it is clear that AAIGA can get a stable average fitness early, and SGA has a more random optimal process, then get partly solution. So, Adaptive strategy with distant kinsman pointer and immunity operator can make GA has better global optimal capability.

6. REFERENCES

[1] J.H. Holland, Adaptation in Natural Artificial Systems, MIT Press, 1975.

[2] J.D. Bagley, The Behavior of Adaptive System Which Employ Genetic and Correlation Algorithms. University of Michigan, 1976.68-7556.

[3] D.E. Goldberg. Computer-aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning. Department of Civil Engineering, University of Michigan, No.8402282, 1983.

[4] M. Scinvivas, L.M. Patnaik, “Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms,” Institute of Electrical and Electronics Engineers Trams. SMC, August 1994, 24(4): 656-666.

[5] X.C. Yan, W.G. An, et al. “Adaptive Immunity Genetic Algorithm,” Chinese Journal of Applied Mechanics. 2005,3:445-448.

An Efficient Strategy for Multi-objective Optimization Problem

Yang Peng
School of Computer and Information Engineering, Zhejiang GongShang University
Hangzhou, Zhejiang 310035, P.R.China
Email: pengyang@mail.zjgsu.edu.cn

ABASTRACT

The location-routing problem is one of the problems in distribution network designing and logistic management, whose mathematical model is proposed in this paper firstly. which is recognized as NP-hard. In this paper, a particle swarm optimization method for Location-Routing problem is introduced. Which adopts a novel particle representation to correspond to the LRP, and modified the objective function with additive penalty function part in order to make the solution more feasible, the Procedure of the PSO algorithm for LRP was then presented. From an illuminated example, the availability and performance of proposed PSO can be approved.

Keywords: Particle swarm optimization; Location- Routing problem; integrated optimization; logistics.

1. INTRODUCTION

The conceptual foundation of LRP studies date back to more than thirty years ago, although those earlier studies are far from capturing the total complexity of LRP, they already recognized the close relation between location and transportation decisions. In practical situation, the location of facilities not only affects locational costs, but also has a major impact on routing costs (Webb [2], Salhi and Rand [3]), the overall problem is, in fact, a multilevel LRP that should be taken jointly both location and routing problems into consideration. Location-Routing Problem (LRP) is one of the problems in integrated logistics optimization. Which is defined to find the optimal number and locations of the facilities, and simultaneously to get the vehicle schedules and distribution routes so as to minimize the total system cost (Tai-His Wu,C,J [1]). In practical situation, the location of facilities not only affects locational costs, but also has a major impact on routing costs(Webb[2], Salhi and Rand [3]), the overall problem is, in fact, a multilevel LRP that should be taken jointly both location and routing problems into consideration, they have been very often solved independently. In the last years, however, several works have addressed LRP of various characteristics. Two sub-problems, LAP (location-allocation problem) and VRP (vehicle routing problem) of LRP have been recognized as NP-hard, thus LRP also belongs to the class of NP-hard problem. Since the exact algorithm seems infeasible to solve the problem, it is worthwhile to develop some heuristic methods. For instance, Nagy and Salhi[4,5] adopted the concept of nested methods for LRP; Tai-His Wu[1] proposes Simulation annealing method which decomposes the LRP into a LAP and a VRP; Maria A S, Juan A [8] also present a two-phase tabu search architecture for the solution of the LRP. Many researchers (Chen TW[6],S.c.Liu,S.B Lee[9], etc.)Resort to iterative heuristic approaches, in which at each iteration a location problem plus a routing problem are solved independently, and Min H,J.[7] synthesizes the past research and suggests some future research directions for the LRP.

This paper proposes a Particle swarm optimization(PSO) method, which is developed by Kennedy and Eberhart [10], PSO is an evolutionary algorithm that simulates the social behavior of bird flocking to a desired place, which follows a collaborative population-based search model, each individual of the population, called a ‘particle’, flies around in a multidimensional search space looking for the optimal solution. Particle, then, may adjust their position according to their own and their neighboring-particles experience, PSO starts with initial solutions and updates them from iteration to iteration. PSO has many advantages over other heuristic techniques such that it can be implemented in a few lines of computer code, it requires only primitive mathematical operators, and it has great capability of escaping local optima. From our survey, ours is the first attempt in proposing a PSO algorithm for the Location-Routing problem.

The remainder of the paper is organized as follows: Section 2 defines the combined multi-facility location-routing problem and gives a corresponding mathematical formulation; Section 3 describes our PSO method for the LRP. Experimental and results are reported in Section 4; Section 5 summarizes the conclusions of this study.

2. PROBLEM DEFINITION AND MATHMATI- CAL MODEL

Location-Routing Problem can be stated as follows. A feasible set of potential facility sites and location and expected demands of each customer are given, which will meet its demands. The shipments of customer demand are carried out by vehicles, which are dispatched from the Facilities and operated on routes that include multiple customers. There is a fixed cost that is associated with opening a facility at each potential site. A distribution cost associated with any routing of vehicles includes the cost of delivery operations. The cost of delivery operations is linear in the total distance the location of the facilities and the vehicle routes from the facilities to customers, the objective is to minimize the sum of the location an distribution costs so that the facilities’ capacities are not exceeded. The hypotheses are as follows: (1)The transportation is just in time. (2)The facility is both starting point and destination of circular vehicle routing; each facility serves more than two customers. (3)Nature of demand/supply is deterministic. (4)There are multiple facilities. (5)Size of vehicle fleets is a single vehicle. (6)Vehicle capacities are determined. The total amount of goods is limited in every route by each vehicles capability. (7)facility capacities are undetermined; not all facilities have been chosen in every decision. (8)Each customer is served by one and only one vehicle. (9)Each facility is considered

as a separate entity, not linked to the other facilities. (10)The objective is to minimize total cost.

Model of above LRP is defined as follows.

$$f(x) = \min \sum_{i\in S}\sum_{j\in S}\sum_{k\in V} C_{ij} X_{i\,jk} + \sum_{r\in G} F_r Z_r \quad (1)$$

subject to

$$\sum_{k\in V}\sum_{i\in S} {}_{ijk} = 1, \quad \forall j \in H \quad (2)$$

$$\sum_{i\in H}\sum_{j\in S} q_j X_{ijk} \le Q_k, \quad \forall k \in V \quad (3)$$

$$\sum_{i\in S} X_{ipk} - \sum_{j\in S} X_{pjk} = 0, \ \forall k \in V, p \in S \quad (4)$$

$$\sum_{r\in G}\sum_{j\in H} X_{ijk} \le 1, \qquad \forall k \in V \quad (5)$$

$$\sum_{k\in V} X_{rjk} + Z_r + Z_j \le 2 \qquad \forall m = 1,\cdots,R, r \in G \quad (6)$$

$$\sum_{k\in V}\sum_{i\in H} X_{rik} - Z_r \ge 0, \forall r \in G \quad (7)$$

$$\sum_{i\in H} X_{rik} - Z_r \le 0, \ \forall k \in V, r \in G \quad (8)$$

Parameters used are as follows:

G={r|r=1,…,m}: the set of m feasible sites of candidate facility;

H={I|i=1,…N}: the set of N customers to be served;

$S = \{G\} \cup \{H\}$:the set of all feasible sites and customers, it is also referred to all nodes;

V={vk|k=1,…,K}: the set of K vehicles available for routing from the facilities;

Cij: average annual cost per distance traveling from node i to node j, $i \in S, j \in S$

Fr: annual cost of establishing and operating a facility at site r(r=1,…,m);

qj: average number of units demands by customer, $j \in H$

Qk: capacity of vehicle k(k=1,…,K);

dij: distance from node i to node j;

Xrmk, Xrjk: traveling by vehicle k from node r to customer m or to customer j, respectively.

In this model, the objective function minimizes the total cost of routing, establishing and operating the facilities. Constraint (2) ensures that each customer is served by one and only one facility. Constraint (3) ensures that the total demands served by one facilities do not surplus its' capacity, while (4) is the route continuity constraints, which implies that the vehicle should leave every point entered by the same vehicle. Constraint (5) guarantees that each vehicle is routed from one facility. Constraint (6) guarantees that there is no link between an two facilitys,

Constraints (7) and (8) require that a vehicle only be from a facility if that facility is opened; xijk and Zr are decision variables, if point i immediately precedes point j on route k($i, j \in I \cup j$) , xijk is equal to 1, otherwise is 0, and if facility r is established, Zr equals to 1, otherwise is 0.

3. REPRESENTAION OF THE PARTICLE

3.1. Fundamental principle of PSO

Like evolutionary algorithm, PSO conducts search using a population (called swarm) of individuals (called particle) that are updated from iteration to iteration. Each particle represents a candidate position (i.e., solution) to the problem at hand, resembling the chromosome of GA. A particle is treated as a point in an M-dimension space, and the status of a particle is characterized by its position and velocity. Initialized with a swarm of random particle, PSO is achieved through particle flying along the trajectory that will be adjusted based on the best experience or position of the one particle(called local best) and the best experience or position ever found by all particles(called global best). The M-dimension position for the ith iteration can be denoted as Xi(t)={xi1(t),xi2(t)…,xiM(t)},similarly, the velocity(i.e.,distance change), also a M-dimension vector, for the ith interationcan be described as Vi(t)={vi1(t),vi2(t),…,viM(t)}, the particle-updating mechanism for particle flying(i.e., search process) can be formulated as follows:

$$V_i(t) = w(t)V_i(t-1) + c_1 r_1 (X_i^L - X_i(t-1)) + c_2 r_2 (X^G - X_i(t-1) \quad (11)$$

$$X_i(t) = V_i(t) + X_i(t-1) \quad (12)$$

Where i=1,2,…,P, and P means the total number of the particles in a swarm, which is called population size; t=1,2,…,T, and T means the iteration limit; $X_i^L = \{x_{i1}^L, x_{i2}^L, \cdots, x_{iM}^L\}$ represents the local best of the ith particle encountered after t-1 iterations, while $X^G = \{x_1^G, x_2^G, \cdots, x_m^G\}$ represents the global best among all the swarm of particles achieved so far. c1 and c2 are positive constants(namely, learning factors), and r1 and r2 are random numbers between 0 and 1; w(t) is the inertia weight used to control the impact of the previous velocities on the current velocity, influencing the trade-off between the global and local experiences.

Formula (11) is used to calculate a particle's new velocity according to its previous velocity and the distances from its current position to its local best and the global best, Formula (12) is used to calculate a particle's new position by utilizing its experience and the best experience of all particles. Formula (11) and (12) also reflect the information-sharing mechanism of PSO.

3.2 the particle representation

In this section, we describe the formulation of a PSO algorithm for the LRP, one of the key issues in designing a successful PSO algorithm is the representation step, i.e. finding a suitable mapping between problem solution and PSO particle. In this paper, we reference the method in [11,12], setup a 2N-dimension search space, N is the number of customer to be served, 2N-dimension particle X can be regarded as two N-dimension vectors: Xv represented the customer is served by a facility, and Xr represents the order in the route of the facility. For example, there are 6 candidate facilities and 10 customers to be served, a particle's position is showed in table 1.

Table 1. The representation of particle's position

customer	1	2	3	4	5	6	7	8	9	10
X_v	3.2	4.1	3.1	4.5	7.2	4.2	6.7	4.7	7.1	6.2
Xr	1.2	3.4	5.6	8.1	2.3	4.6	2.1	6.2	4.8	8.1

Then we should turn the Xv into integer, and if

Xvi=Xvj(i,j=1,2,…10), the order should be sorted according to Xr, so the particle can be converted as table 2.

Table 2. The updated representation

customer	1	2	3	4	5	6	7	8	9	10
X_v	3	4	3	4	7	4	6	4	7	6
Xr	1	1	2	4	1	2	1	3	2	2

The particle represents a solution, the selected facility and the related routes are showed in table 3, where 0 denotes the facility itself, which is the begin and the end of the route.

Table 3. The solution for the problem

selected facility	route for the facility
3	0−1−3−0
4	0-2-6-8-4-0
6	0-7-2-0
7	0-5-2-0

This representation makes each customer can be served by only one facility, and number and amount of selected facility also can be embodied. Though number of dimension is increased, PSO shows the good characteristic in multi-dimension optimization, the computation complication does not increase.

3.3 the modified objective function

LRP is a restricted combinatorial optimiza- tion problem, formulation 3 scripts the restriction of route capacity, that is, the sum of demands to be served of one location can not surplus its' capacity. This paper adopts a modified objective function that includes a penalty term associated with infeasible, R is a large number as penalty weigh, which multiple the overload, and then is added to the objective function. The modified objective function as follows:

$$F(x) = \min(\sum_{i\in S}\sum_{j\in S}\sum_{k\in V} C_{ij} X_{i\,jk} + \sum_{r\in G} F_r Z_r + R.\sum_{k}\max(\sum_{i}\sum_{j} C_{ij} X_{ijk} - q_k, 0) \quad (13)$$

3.4 Procedure of the PSO algorithm for LRP

The procedure of the PSO can be stated as follow:

Step1: initialize the particle swarm,

1) For each 2N-dimension particle in the population, the position vector Xv={X1,X2,…,XN}, each element is randomly generated from 1~M, and Xr={XN+1,XN+2,…,X2N} randomly from 1~N;

2) Each velocity vector is also 2N-dimension, Vv={V1,V2,…,VN}, each element is randomly generated from -M~M, and Vr={VN+1,VN+2,…,V2N} randomly from -N~N;

3) Evaluate the fitness of each particle in the population, the fitness function is described as formula (13)

4) Initialize the Gbest with the best fitness (lowest cost) among the population.

5) Initialize the Pbest with a copy the population;

Step2: Repeat until the number of generation equal to maximum generation number or satisfy the stop situation.

1) For each particle, update X and V according to equation (2) and equation (3), the updated particle position must be subject to the limit[Xmin,Xmax]=[1,M], each element of the 2N-dimension particle that is beyond [Xmin,Xmax] can be adjusted as: if Xi(t)>Xmax, then Xi(t)=Xmax ; else if Xi(t)<Xmin, then Xi(t)=Xmin; and each element of the 2N-dimension particle velocity that is beyond [-Vmax,Vmin]=[-N,N] should be adjusted as: if Vi(t)>Vmax then Vi(t)=Vmax, else if Vj(t)<-Vmax, then Vj(t)=-Vmax.

2) Turn the 2N-dimension particle X into appropriated representation as section 3.2.

3) Evaluate the fitness of all particle in the new population;

4) update the Gbest, if one particle's fitness is better than current Gbest; for each particle, if the fitness is better than its Pbest, then update it;

Step 3: Stop the PSO and get the approximately optimal solution

The PSO execution should be terminated if the current iteration meets any one of the stop signals. The stop signals considered here include: (1) maximum number of iterations with steady global best (i.e., maximum number of iterations since last updating of the global best) and (2) maximum total number of iterations. Then the optimal solution to LRP has been get, which corresponds to the global best particle-representation.

4. COMPUTATIONAL EXPERIMENTS

Numerical experiments were conducted to examine the computational effectiveness and efficiency of our proposed method. The heuristic methods are coded using the Visual C++ programming language and the tests are carried out on a PC Pentium 1.8GHZ.

The demand for each customer in any test is randomly selected from a uniform distribution U[450, 600] for each month. The demand during lead time for each customer is randomly selected from a uniform distribution U [0, 10]. The location (x, y) of each customer and candidate depot is randomly selected from a uniform distribution U[0, 100]. In our PSO, the population size is 60,c1 and c2 are set to 1 and 1.5 respectively, and penalty coefficient R is 108, maximum iteration is 50.

number of depots	number of customers	Tabu Search cost	Tabu Search cpu	Based PSO cost	Based PSO cpu
10	100	79542	8	79238	9
10	150	121564	10	110256	10
10	180	156325	11	152357	11
10	200	162321	13	160241	14
10	220	173243	14	172314	14
20	100	67356	7	67135	7
20	150	102325	9	110234	8
20	180	121325	11	121034	10
20	200	134241	12	128967	12
20	250	142315	14	147782	15
25	200	124531	14	122215	14
25	220	131025	15	130021	16
25	250	150124	15	145781	15
25	300	162437	17	161325	17
25	350	180219	19	179324	18

Table 4. Results for the problem and comparison.

Table 4.1. show the average solutions and average CPU times for this proposed two- phase PSO based method and other heuristic method described earlier, which is presented in literature 13. It is found that our method is better than Tabu-Search based method in total costs. And also, when the number of candidate depots increases, the total system

costs for all the two solutions decrease. When the number of customers increases, the total system costs for all the three heuristic solutions increase. However, in terms of average CPU time, these two methods are efficient, for all average CPU times in Table 1 are less than 30 seconds.

5. CONCLUSION

In this paper, a particle swarm optimization method for Location-Routing problem is proposed. It adopts novel particle representation to correspond to the integrated optimization problem in logistics, and modifies the objective function with additive penalty function part in order to get the feasible solution. The proposed method can solve the special LRP, which is under many assumed situations. From the illustrational example, the availability and performance of proposed PSO can be proved. Further related research directions will be to develop more complex model of LRP such as the multi- product multi-depot location -routing problem and those solution algorithm.

REFERENCES

[1] T.H.Wu, C.Y.Low, et al. “Heuristic solution to multi-facility location-routing problems”. Computers &operations Research. 2002,1393-1415.

[2] Webb MHJ. “Cost functions in the location of facility for multiple delivery journal Research Quarterly”. 1968,11-20.

[3] Salhi S,Rand GK. “The effect of ignoring routes when locating facilitys”. European Journal of Operational Research.1989, 11-20.

[4] G. Nagy, S.Salhi, “Nested heuristic methods for the location-routing problem”. Journal of operational Research Society.1996,66-74.

[5] G. Nagy, S.Salhi, “A nested location-routing heuristic using route length estimation”. Studies in Locational Analysis. 1996;10:109-27.

[6] T.W.Chien, “Heuristic procedures for practical-sized uncapacitated location-capcitated routing problem”. Decision Sciences. 1993,995-1021.

[7] Min H,Jayaraman V, et al. “Combined location-routing problem:a synthesis and future research directions”. European Journal of Operational Research. 1998,1-15.

[8] Maria Albareda-Sambola, Juan A, Diaz, Elena Fernandez. location-routing problem. Computer & Operations Research. 2005,407-428.

[9] S.C.Liu, S.B.Lee. “A two-phase heuristic method for the muti-facility location routing problem taking inventory control decisions into consideration”. Int J Adv Manuf Technol. 2003,941-950.

[10] J. Kennedy, R.C. Eberhart, “Particle swarm optimization”. Proceedings of the IEEE International Conference on Neural Networks.(1995)1942-1948.

[11] R.C. Eberhart, Y. Shi, “Particle swarm optimization: developments, applications and resources”. Proc .Congress on evolutionary computation, 2001, Seoul, Korea.P.81-6.

[12] A. Salman,I. Ahmad, et al. “Partical swarm optimizaion for task assignment problem”. Microprocessors and Microsystems. 2002,26:363-371

[13] D. Tuzun, L.I. Burke. “A two- phase tabu search approach to the location routing problem”. European Journal of Operational Research.1999, 87-99

A Hybrid Quantum Genetic Algorithm Based On Clonal Selection Principle

Bin Han1,2 Xin Zuo2 Qianqian Luo2 Bin Su2 Shitong Wang1
1. School of Information, Southern Yangtse University,
WuXi, Jiangsu, 214036, China
Email: cnhanbin@tom.com
2. School of Electronics and Information,JiangSu University of Science and Technology,
ZhenJiang, Jiangsu, 212003, China;
Email: zx1005@tom.com

ABSTRACT

A hybrid quantum genetic algorithm (HQGA) is presented in this paper, based on the concept and principle of clonal selection. In the framework of this proposed algorithm, quantum genetic procedure is used to find coarse solutions. The clone selection and affinity maturation process procedure is used to find the finer solution. The power of this proposed algorithm comes from the fact that it embodies that evolutionary computing with the qubit representation has a better diversity characteristic and that the clonal selection and affinity maturation process enhances the solution quality. Several examples are demonstrated here to confirm the above claims.

Keywords: Quantum Genetic, Clonal Selection Principle, Hybrid Algorithm.

1. INTRODUCTION

Genetic programming (GP) is one of the most useful, general-purpose problem solving techniques. It has been used to solve a wide range of problems, such as symbolic regression, data mining, optimization, and emergent behavior in biological communities. In many ways, genetic algorithms and their extension offer an outstanding combination of flexibility, robustness and simplicity.

Quantum genetic algorithm (QGA) is a novel genetic algorithm, which is based on the concept and principle of quantum computing such as qubits and superposition of states. Instead of binary, numeric or symbolic representation, by adopting qubit chromosome as a representation, QGA can represent a linear superposition of solutions due to its probabilistic representation. QGA has an excellent capability of global search due to its diversity caused by the probabilistic representation, and it can achieve better solutions than CGA's in a short time [1].

The most notable work in this field is due to Han and Kim [1, 2, 3, 4] who devised a new evolutionary approach to solve combinatorial optimization problems. In the latest work they gave an efficient approach that can tune the parameters finely in QGA. At the same time, they developed a parallel version of QGA. Moreover, several quantum inspired learning algorithms like Quantum Inspired Genetic Algorithms by Narayanan and Moore were recently proposed [5].

However, there are still some insufficiencies; Indeed, one of the major problems usually associated with the use of QGAs is the premature convergence to solutions coding local optima of the objective function.

The clonal selection principle is used to explain the basic features of an adaptive immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigens are selected to proliferate. The selected cells are subject to an affinity maturation process, which improves their affinity to the selective antigens [6, 7, 8, 9]. In fact, the clonal selection can yield a group of the mutation solutions around a candidate solution according to its affinity, in order to increase the chances of escaping from local optimal solutions [10].

In this paper, a hybrid quantum genetic algorithm (HQGA) is presented based on the quantum genetic algorithm and the clonal selection principle. Its advantage exists in that it can enhance the solution quality. In other words, the chance that the obtained solutions approximate to the optimal solution is greatly increased.

The paper is organized as follows. Section 2 gives a brief overview of the quantum genetic algorithm. Section 3 reviews the clonal selection algorithms. They serve as the basis for explaining the Improved Quantum Genetic Algorithm. Section 4 gives the Improved Quantum Genetic Algorithm. In Section 5, the Hybrid Quantum Genetic Algorithm is experimentally compared with the Standard Quantum Genetic Algorithm. Section 6 concludes the paper and discusses future work.

2. QUANTUM GENETIC ALGORITHM

In this section, QGA proposed in [1] is briefly described.

2.1 Representation

GQA is based on the concepts of qubits and superposition of states of quantum mechanics. The smallest unit of information stored in a two-state quantum computer is called a quantum bit or qubit . A qubit may be in the '1' state, in the '0' state, or in any superposition of the two. The state of a qubit can be represented as

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

Where α and β are complex numbers that specify the probability amplitudes of the corresponding states. $|\alpha|^2$ gives the probability that the qubit will be found in '0' state and $|\beta|^2$ gives the probability that the qubit will be found in the '1' state. Normalization of the state to unity guarantees

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2)$$

If there is a system of m-qubits, the system can represent 2^m states at the same time. However, in the act of observing a quantum state, it collapses to a single state.

A Q-bit individual as a string of m-bits is defined as

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_m \end{bmatrix} \quad (3)$$

Where $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1,2,...,m$. Q-bit representation has the advantage that it is able to represent a linear superposition of states in probabilistical way. The Q-bit representation has a better characteristic of generating diversity in population than any other representations.

QGA is a probabilistic algorithm similar to other GAs. QGA, however, maintains a population of Q-bit individuals, $Q(t) = \{q_1^t, q_2^t, ...q_n^t\}$ at generation t, where n is the size of population, and q_j^t is a Q-bit individual defined as

$$q_j^t = \begin{bmatrix} \alpha_{j1}^t & \alpha_{j2}^t & \alpha_{j3}^t & \cdots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \beta_{j3}^t & \cdots & \beta_{jm}^t \end{bmatrix} \quad (4)$$

where m is the number of Q-bits, i.e., the string length of the Q-bit individual, and $j = 1,2,...,n$.

2.2 QGA

Here, we state the details of QGA.

Procedure QGA

Begin
t ←0
1) initialize Q(t)
2) make P(t) by observing the states of Q(t)
3) evaluate P(t)
4) store the best solution among P(t) into B(t)
5) **while (not termination condition) do**
Begin
t ← t+1
6) make P(t) by observing the states of Q(t-1)
7) evaluate (t)
8) update Q(t) using Q-gates
9) store the best solution among B(t-1) and P(t) into B(t)
End
End

In the while loop, binary solutions in P(t) are formed by observing the states of Q(t-1) as in step (2), and each binary solution is evaluated for the fitness value. It should be noted that x_j^t in P (t) can be formed by multiple observations of $q^{t-1}{}_j$ in Q (t-1).

In the step (8), Q-bit individuals in Q(t) are updated by applying Q-gates defined as a variation operator of QGA, by which operation the updated Q-bit should satisfy the normalization condition, $|\alpha'|^2 + |\beta'|^2 = 1$, where α' and β' are the values of the updated Q-bit. The following rotation gate is used as a basic Q-gate in QGA, such as

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \quad (5)$$

where $\Delta\theta_i$, $i = 1,2,...,m$, is a rotation angle of each Q-bit toward either 0 or 1 state depending on its sign.

The magnitude of $\Delta\theta_i$ has an effect on the speed of convergence, if it is too big, the solutions may diverge or converge prematurely to a local optimum. If the value of rotation angle is smaller, the convergence speed may be changed to be slower. In our work, we propose a method based on the clonal selection principle to overcome this shortcoming.

3. THE CLONAL SELECTION THEORY

When an animal is exposed to an antigen, some subpopulation of its bone marrow derived cells (B lymphocytes) respond by producing antibodies (**Ab**). Each cell creates only one kind of antibody, which is relatively specific for the antigen. By binding to these antibodies (receptors), and with a second signal from accessory cells, such as the T-helper cell, the antigen stimulates the B cell to proliferate (divide) and mature into terminal (non-dividing) antibody secreting cells, called plasma cells. The various cell divisions (mitosis) generate a clone, i.e., a set of cells that are the progeny of a single cell. While plasma cells are the most active antibody secretors, large B-lymphocytes, which divide rapidly, also secrete **Ab**, albeit at a lower rate. While B cells secrete **Ab**, T cells play a central role in the regulation of the B cell response and are preeminent in cell-mediated immune responses.

The clonal selection algorithm (CAS) proposed in [6,7] can be described as follows:

Procedure CAS

•**Ab**: available antibody repertoire (**Ab**$\in S^{N\times L}$, **Ab**=**Ab**{r} ∪ **Ab**{m});
•**Ab**{m}: memory antibody repertoire (**Ab**{m} $\in S^{m\times L}$, $m \le N$);
•**Ab**{r}: remaining antibody repertoire (**Ab**{r} $\in S^{r\times L}$, $r = N - m$);
•**Ag**{M}: population of antigens to be recognized (**Ag**{M} $\in S^{M\times L}$);
• ***fj***: vector containing the affinity of all antibodies with relation to antigen ***Agj***;
•**Ab**$^j{}_{\{n\}}$: n antibodies from **Ab** with highest affinities to ***Agj*** (**Ab**$^j{}_{\{n\}} \in S^{n\times L}$, $n \le N$);
•**C** ^{j}: population of N_c clones generated from **Ab**$^j{}_{\{n\}}$ (**C** $^j \in S^{Nc\times L}$);
•**C** ^{j*}: population **C** ^{j} after the affinity maturation (hypermutation) process;
•**Ab**{d}: set of d new molecules that will replace d low-affinity antibodies from **Ab**{r}(**Ab**{d} $\in S^{d\times L}$, $d \le r$);
•***Ab*** $^*_j$: candidate, from **C***j**, to enter the pool of memory antibodies.

Begin

1) Randomly choose an antigen ***Agj*** (***Agj*** ∈ **Ag**) and present it to all antibodies in the repertoire **Ab**= **Ab**{r} ∪ **Ab**{m} ($r + m = N$);

2) Determine the vector ***fj*** that contains its affinity to all the N antibodies in **Ab**;

3) Select the n highest affinity antibodies from **Ab** composing a new set **Ab**^{j}{n} of high affinity antibodies with relation to ***Agj***;

4) The n selected antibodies will be cloned (reproduced) independently and proportionally to their antigenic affinities, generating a repertoire $\boldsymbol{C}^j$ of clones: the higher the antigenic affinity, the higher the number of clones generated for each of the n selected antibodies;

5) The repertoire $\boldsymbol{C}^j$ is submitted to an affinity maturation process inversely proportional to the antigenic affinity, generating a population $\boldsymbol{C}^{j*}$ of matured clones: the higher the affinity, the smaller the mutation rate;

6) Determine the affinity fj^* of the matured clones $\boldsymbol{C}^{j*}$ with relation to antigen ***Agj***;
7) From this set of mature clones $\boldsymbol{C}^{j*}$, re-select the highest affinity one ($\boldsymbol{Ab}_j^*$) with relation to ***Agj*** to be a candidate to enter the set of memory antibodies $\mathbf{Ab}\{m\}$. If the antigenic affinity of this antibody with relation to ***Agj*** is larger than its respective memory antibody, then $\boldsymbol{Ab}_j^*$ will replace this memory antibody;
8) Finally, replace the *d* lowest affinity antibodies from $\mathbf{Ab}\{r\}$, with relation to ***Agj***, by new individuals.
End

4. THE PROPOSED ALGORITHM

Our hybrid algorithm HQGA integrates the above quantum genetic algorithm QGA with the clonal selection algorithm CSA. The idea behind HQGA can be stated as follows. As we may know, the affinities of antibodies with relation to antigen in the CAS can be seen as the fitness in QGA. In the framework of algorithm HQGA, we use a representation that is based on the concept of qubits to encode the solutions onto chromosomes. Such a representation inherits the advantage that it is able to represent any superposition of states. We first utilize QGA to obtain coarse solutions comparable to the optimal solution, and then it obtains the finer solutions by using CAS. Clonal selection and affinity maturation process in CAS assumes that the capability of searching better local solutions than QGA is enhanced. Algorithm HQGA updates its population using the quantum gates. Evolutionary computing with the qubit representation has a better characteristic diversity.

The proposed algorithm HQGA is described as follows:

Procedure HQGA

•Q(t) is population using Q-bit representation at generation *t*.
• P(t) is the set of the real number solutions at generation *t*. It is produced from Q(t).
• b(t) is the best solution at generation *t*.
• B(t) is the set of the *n* best solution at generation *t*.
• C(t) is the set of clones.
• C*(t) is generated from C(t) by mutation.

Begin
t ←0
1) initialize Q(t)
2) make P(t) by observing the states of Q(t)
3) evaluate P(t)
4) select and store the *n* best solutions among P(t) into B(t);
5) store the best solution b(t) among B(t)
6) **Do while** (**not termination condition)**
Begin
t ← t+1
7) make P(t) by observing the states of Q(t-1)
8) evaluate P(t)
9) update Q(t) using quantum gates U(t)
10) if the best solution in P(t) and B(t-1) is better than b(t), store it into b(t)
11) select the *m* best solutions among P(t) and B(t-1)
12) The *m* selected individuals will be cloned (reproduced) independently and proportionally to their fitness, generating a C(t) of clones: the higher the fitness, the higher the number of clones generated for each of the *m* selected individuals;
13) The C(t) is submitted to an affinity maturation process inversely proportional to the antigenic affinity(fitness), generating a population C*(t) of matured clones: the higher the affinity(fitness), the smaller the mutation rate;
14) evaluate C*(t)
15) From this set of mature clones C*(t) , re-select the *d* highest affinity (fitness) solutions to replace the *d* lower affinity (fitness) solutions in B(t-1) .
16) **if** the best solution in B(t) is better than b(t), store it into b(t)
End
End

In step (13), the affinity maturation process (hypermutation) uses Gaussian mutation, as follows:

$$x' = x + \alpha N(0,1) \tag{6}$$

$$\alpha = (1/\beta)\exp(-f^*) \tag{7}$$

Where, x' is the individual after Gaussian mutation; x is the original individual; α is the mutation rate to each individual; f^* is the antigenic affinity (fitness) of each individual; N (0, 1) is a random Gaussian variable with zero mean and unitary standard deviation.

5. EXPERIMENT SOLUTIONS

In order to validate our approach here, we used several benchmarking functions in [11,12]. In algorithm HQGA, we set the following parameters: Population size=50 , maximum number of iterations=500, number of clones=10.

Example 1:

Maximize: f(x)= $10+\dfrac{\sin(1/x)}{(x-0.16)^2+0.1}$,

where: 0. 01$\leqslant x \leqslant$1； The maximum is 19.8949.

Example 2:

Minimize: $f(x)=x^2+y^2$

where: -5$\leqslant x, y \leqslant$5； The minimum is 0;

Example 3:

Minimize: $f(x)=0.5+\dfrac{\sin^2\sqrt{x^2+y^2}-0.5}{\left[1.0+0.001(x^2+y^2)\right]}$,

where: -100$\leqslant x, y \leqslant$100； The minimum is 0;

The comparative results of HQGA, NQGA and IQGA are shown in Tables 1-3. The solution qualities of HQGA are better than IQGA and NQGA. Our algorithm HQGA achieves the optimal solutions only using 87, 15 and 321 iterations averagely, much less than those of both IQGA and NQGA; Another fact should also be stated here. When we run these algorithms twenty times for this example, the success rate of HQGA is 100%. However, HQGA always outperforms both IQGA and NQGA in the obtained optimal solutions.

Table 1. Results on example 1

Function 1	QGA[11]	NQGA[11]	HQGA
Worst			19.8949
Best	19.8949	19.8949	19.8949
Mean			19.8949
Number of iterations	177.17	139	87
Success rate	87%	100%	100%

Table 2. Results on example 2

Function 2	IQGA[12]	NQGA[12]	HQGA
Best	0.00004587	0.00008108	0.000039261
Mean	0.0005263	0.000527	0.0005066
Number of iterations	102.3000	17.8	15
Success rate	100%	100%	100%

Table 3. Results on example 3

Function 3	IQGA[12]	NQGA[12]	HQGA
Best	0.003874521	0.000018627	0.0000127048
Mean	0.00889486	0.00024215	0.00023638
Number of iterations	952.540	329.127	321
Success rate	69%	100%	100%

6. CONCLUSION AND FUTURE WORK

In this paper, the hybrid quantum genetic algorithm HQGA is proposed. As we may know, evolutionary computing with the qubit representation has a better diversity characteristic diversity than the conventional GA. Clonal selection and affinity maturation process in algorithm CSA outperforms QGA in searching the optimal solution in local area. Our algorithm HQGA integrates these advantages together to achieve better solutions than both QGA and CAS. Our experimental results here confirm its effectiveness. Future work includes the theoretical analysis for algorithm HQGA.

REFERENCES

[1] K.H. Han,J.H. Kim, Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem, IEEE Proc Of the 2000 Congress on Evolutionary Computation, San Diego, USA, IEEE Press, 2000,pp.1354-1360

[2] K.H. Han,J.H. Kim,On Setting the Parameters of Quantum-inspired volutionary Algorithm for Practical Applications ， Lecture Notes in Computer Science (GECCO 2003), eds. E. Cantu-Paz et al., Berlin Heidelberg: Springer-Verlag, Vol. 2723, July 2003,pp. 427-428.

[3] K.H. Han, J.H. Kim, Analysis of Quantum-Inspired Evolutionary Algorithm. 2001 International Conference on Artificial Intelligence, 2, CSREA Press, June 2001. Pp.727-730.

[4] K.H. Han,J.H. Kim,Quantum-inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Transactions on Evolutionary Computation, vol.6, Dec 2002, pp.580-593.

[5] Narayanan, Ajit Mark Moore,Quantum Inspired Genetic Algorithms .International Conference on Evolutionary Computation, 1996,pp.61-66.

[6] Leandro N. de Castro,Fernando J. Von Zuben, Learning and Optimization Using the Clonal Selection Principle ,IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems, vol. 6, n. 3, 2002,pp. 239-251.

[7] Leandro N. de Castro, Fernando J. Von Zuben ,The Clonal Selection Algorithm with Engineering Applications, Proceeding of Artificial Immune System Workshop, Genetic and Evolutionary Computation Conference (GECCO' 2000), 2000. pp.36-37.

[8] Felipe Campelo, Hajime Igarashi, A Clonal Selection Algorithm for Optimization in Electromagnetics , IEEE Transactions on Magnetics , Vol. 41, No. 5, May 2005, pp.1736-1740

[9] L.C.Jiao,H.F.Du, Development and Prospect of the Artificial Immune System, ACTA ELECTRONICA SINICA(In Chinese), 20O3(10), pp.1540-1548

[10] X.R.Zhang,L.C.Jiao, Feature Selection Based on Immune Clonal Selection Algorithm, Journal of Fudan University(Natural Science, In Chinese), Vo1.43, NO.5, 0Ct.2004, pp.926-929

[11] G.X.Zhang,W.D.Jin. Feature selection algorithm based on quantum genetic algorithm. Control Theory& Application (In Chinese). Vol.22, NO.5, 2005.5(22), pp.810-819

[12] G.X.Zhang,N.Li, A Novel Quantum Genetic Algorithm an d Its Application ,ACTA ELECTRONICA SINICA(In Chinese), Vol.32, NO.3 2004,pp.476-479

An Ant Colony Algorithm for Solving Set-Covering Problems

Yang Gao
The School of Information Technology, Southern YangTze University
Wuxi, Jiangshu , People's Republic of China
Email:gy_bison@163.com
Hongwei Ge
The School of Information Technology, Southern YangTze University
Wuxi, Jiangshu , People's Republic of China
Email: ghwjm@sina.com.cn

ABSTRACT

Ant colony algorithm is an optimized model based on the principle of swarm intelligence. In this paper, we use ant colony algorithm to find the optimization solution of the set-covering problems. In order to speed up the convergence of algorithm, the improved penalty function is used in the process of pheromone update. The experimental results prove that using the ant colony algorithm to solve the problem of set covering is efficient and feasible.

Keywords: Ant Colony Algorithm, Colony Intelligence, Set-covering, NP-complete Problem.

1. INTRODUCTION

Set-covering problem is a well-known NP-hard optimization problem, which usually cannot be solved exactly within a reasonable amount of time. Many real-world problems can be cast as a Set Covering Problem, for example, crew scheduling in airlines and driver scheduling in public mass transport. These problems are often large in size, and we need to use some kinds of heuristic techniques to find the approximate solutions. In recent years, some general-purpose heuristics, such as genetic algorithm and neural network algorithms, have already been used to solve NP-hard problems. Since 1996, a new type of heuristic named ant colony optimization (ACO) heuristic, proposed by M.Dorigo et al. [1], has gradually attracted much attention for its specific natural metaphor and learning mechanism. In this paper, ACO has been used to solve the set-covering problems successfully, and the experimental results also prove that our method is efficient.

2. SET-COVERING PROBLEM (SCP)

The set-covering problem can be defined as follows. Given a set U of elements and a set S of subset t_i of U, $i = 1 \cdots |S|$ ($|S|$ is the cardinality of S), find the smallest subset $T \subset S$ that contains all elements of U :

$$\bigcup_{t_i \in T} t_i = U$$

The set-covering problem has been proven to be a NP-complete [2] problem that was studied extensively in the literature and is a model for several important applications such as crew or railway scheduling. In recent years, many competitive solutions have been proposed, most of which based on heuristic approaches [3, 4, 5, 6]. In this paper, we study the ant colony optimization for the SCP, and the experimental results show that improved ACO algorithm is better than SCHF algorithm [6], which is a recent and highly effective set-covering algorithm.

3. THE BASIC CONCEPTS OF ANT COLONY OPTIMIZATION

The ACO heuristic has been inspired by the observation on real ant colony's foraging behavior, and on that ants can often find the shortest path between food source and their nest. The principle of this phenomenon is that ants deposit a chemical substance (called pheromone) on the ground; thus, they mark a path by the pheromone trail. An ant encountering a previously laid trail can detect the dense of pheromone trail. It decides with high probability to follow a shortest path, and reinforce the trail with its own pheromone. The larger amount of the pheromone is on a particular path, the larger probability is that an ant selects that path and the path's pheromone trail will become denser. At last, the ant colony collectively marks the shortest path, which has the largest pheromone amount. Such simple indirect communication way among ants embodies actually a kind of collective learning mechanism.

4. ACO HEURISTIC FOR SET-COVERING PROBLEM

In ACO algorithm, an approximate approach should be used to simulate the ant's foraging process. Generally, a stochastic greedy approach is adopted. For set-covering problem, the solution construction procedure of ants is described as follows.

An ant initially starts from a randomly selected set t_i , let $D = \{t_i\}$, $U = U - t_i$, $S = S - t_i$. The set D is called feasible set list. At each construction step, it selects next set t_j from S with a probability distribution given by

$$P_j^k = \begin{cases} \arg\max \left\{ \left\lceil \tau_j^{\alpha} \right\rceil \cdot \left\lceil \eta_j^{\beta} \right\rceil \right\} & , \text{ if } q \le q_0 \\ \dfrac{\tau_j^{\alpha} \eta_j^{\beta}}{\sum_{h \in S} \tau_h^{\alpha} \eta_h^{\beta}} & , \text{Otherwise} \end{cases} \quad (1)$$

where η_j is a function that assigns to each valid component —possibly depending on the current construction step—a heuristic value which is also called the heuristic information. τ_j is called pheromone trail . q is a uniform random variable in [0, 1]. q_0 has been studied in [7] and it was found that $q_0 = 0.5$ is a good choice. The value of parameters of α and β, $\alpha > 0$ and $\beta > 0$, determines the relative importance of pheromone value and heuristic information. In this paper we set $\alpha = \beta = 1$. Suppose t_j be selected, then $D \cup t_j \to D$, and U is recomputed. This single step is repeated until U is empty. Finally a covering set is obtained.

Heuristic information η_j: Heuristic information η_j, which is induced from problem itself, reflects in what degree the set S_j would belong to an optimal solution. Given any sub-set $t_i \in T$, let $|t_i|$ be the number of elements incident on t_i. By the definition of a covering set, it is obviously that the higher order $|t_i|$ a subset has, the more probable it is in a covering set. So $1/|t_i|$ can be used as heuristic to produce an approximate solution through greedy approach. But computational experience showed that this is not a good choice for SCP problem when combined with ACO algorithm. In this paper, we choose SCHF [6] function as η_j, and compute the heuristic information according to

$$\eta_i = \frac{1 + R(t_i)}{P(t_i) - 1} + \frac{1}{N - |t_i|} \tag{2}$$

Where $N = |S|$, $P(t_i)$ and $R(t_i)$ are called covering degree of t_i and certain-choosing degree of t_i respectively and their calculation are same as[6].

Pheromone Update: The pheromone trail τ_j is the core of ACO learning mechanism, and its value varies with time to record the probability of each set in an optimal solution. In this way, new and promising areas of the search space can be explored. Here, the pheromone update rule is modified according to the following equation

$$\tau_j(t+1) = \left(1-\rho\right)\tau_j(t) + \sum \Delta\tau_j^k - \sigma \max\left(\tau_j - \tau_{\min}, 0\right)^2 \tag{3}$$

Where

$$\Delta\tau_j^k = \begin{cases} Q/|D| & \text{, If ant k choose the set } t_j \\ 0 & \text{, otherwise} \end{cases}$$

$|D|$ is the cardinality of set found by ant k, Q is a positive constant. The parameter $\rho \in (0,1]$ is called evaporation rate. It has the function of uniformly decreasing all the pheromone values. σ is the penalty coefficient. In this paper, we let $\sigma = e^{-\frac{1}{T}}$ so that algorithm can eliminate the subset which not belongs to optimal solution obviously and speed up the convergence. τ_j should be changed with iterative number T. On the other hand, in order to avoid the value τ_i increasing too large and prevent the algorithm from stagnation, we give the pheromone update rule as following

$$\tau_j = \begin{cases} \tau_{\max} & \text{, if } \tau_j \geq \tau_{\max} \\ \text{Eq. (3)} & \text{, otherwise} \end{cases} \tag{4}$$

The effect of $\tau_{\min}$, $\tau_{\max}$ has been studied in [4] and this paper get the value of $\tau_{\min}$ and $\tau_{\max}$ according to the experiments.

Improved ACO Algorithm: In summary, our algorithm can be formally stated as follows.

Initialize
While (termination-criterion-not-satisfied)
For k=1 to m
Choose a start set t_i, $D = \{t_i\}$
Repeat
Select an item t_j *with probability* P_j^k *given by Eq. (1) and update* D, U
Until U *is empty*
End
Save the best solution found so far
Update the pheromone level on all set according to Eq. (4)
End

5. EXPERIMENTAL STUDIES

In this section, for improved ACO, we will arrange three experiments. In these experiments, the parameters of the improved ACO heuristic are selected by the following way: The initial values of τ_i are equal to one, the number of ants is selected to be 100. The parameter Q should be small enough so that $\Delta\tau_i$ increase slowly and prevent the system from stagnation. In our experiments, we let $Q = 0.05, \rho = 0.8$. The elements of subset in our experiments are a uniform random variable in [1,100].

Experiment1: In order to compare the improved ACO and standard ACO, we implement standard ACO also. The

parameters of standard ACO are selected the same as the parameters of the improved ACO. The data set is generated by the following way:

We generate 100 subsets t_i, the cardinalities of which are a uniform random variable in [1,100]. We let $\bigcup_{i=1}^{100} t_i = U$ as a covered set.

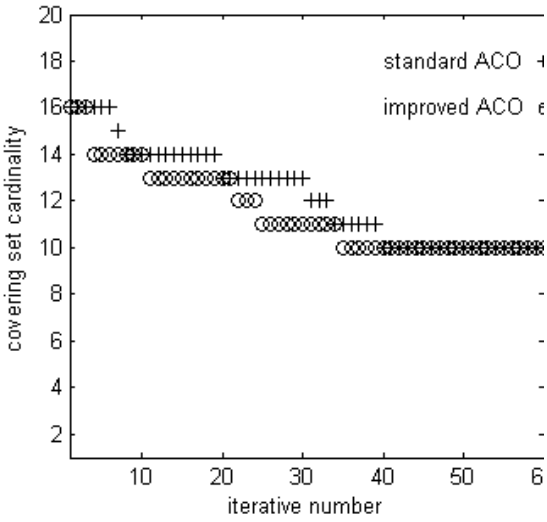


Fig. 1. The comparison of our improved ACO and standard ACO

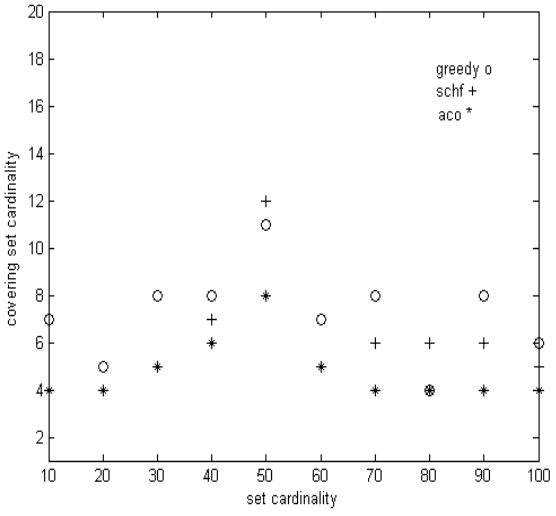


Fig. 2 The Comparison of Optimization Ability of Improved ACO Algorithm, SCHF Algorithm and GREEDY algorithm

Fig. 1 demonstrates the comparison of standard ACO and our improved ACO. It is obviously that improved ACO speed up the convergence of algorithm considerably.

Experiment2: Set cardinality $|S| = 10,20\cdots100$ respectively. We compare our improved ACO with SCHF algorithm and GREEDY algorithm. Because the optimal solution of our problem is unknown, the iterative number is selected to be 80 as a terminal criterion. Also the algorithm is ended when it does not generate better solutions in 30 sequent iterations.

The comparison of three algorithms is shown in Fig.2, from which we can easily find that the global optimization ability of the improved ACO is the best one. This experiment proves that using the ant colony algorithm to solve the problem of set covering is more desirable.

Experiment 3: ACO is an algorithm based on probability. In this experiment, the iterative number is selected to be 30, 80 and 100 respectively, and set cardinality $|S|$ is selected to be $20,40,\cdots,100$ respectively. For each case, we execute the improved ACO algorithm 100 times. Table1 shows the variance of solutions in different iterative number and different set cardinality, from which we can choose the better iterative number according to the practical problem.

Table 1. The variance of solutions in different iterative number and different set cardinality

iterative number \ set cardinality	20	40	60	80	100
30	0.22	0.94	0.9475	0.91	0.84
80	0	0.25	0.31	0.35	0.54
100	0	0	0	0.04	0.2

6. CONCLUSION

ACO heuristic is a new simulation algorithm. In the process of solving the set-covering problem, we find that the ant colony algorithm can always find a better solution. Although ant colony algorithm has been successfully applied to various problems of optimization, it still has the problem of long searching time. How to shorten the searching time is worthy to be studied in the future.

7. REFERENCE

[1] M. Dorigo, V. Maniezzo, et al, "Ant system: optimization by a colony of cooperating agents", IEEE Trans on Systems, man, and Cybernetics-Part B: Cybernetics, 1996, 26 (1): 29-41.

[2] M. Garey, D. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, San Francisco: W.H.Freeman, 1979.

[3] V. Chvátal, "A greedy heuristic for the set-covering problem. Mathematics of Operations Research", 1979, 4(3): 233-235.

[4] J.P .Marques-Silva, "Integer Programming Models for Optmization Problems in Test Generation", Proceedings of the IEEE Asian-south Pacific Design Automation Conference, February 1998, 481-487.

[5] U. Feige, "A threshold of ln n for approximating set cover", Journal of the ACM, July 1998, 45(4): 634-652.

[6] G.R. Quan, B.R. Hong, et al, "A Heuristic Function Algorithm for Minimum Set-covering Problem", Journal of Software, 1998, 9(2).

[7] T. Stutzle, H.H. Hoos, "MAX-MIN ant system", Future Generation Computer System, 2000, 16(8): 889-914.

Humanoid Grid Management Model and its Implementation Frame

Shen Jiquan1,2 Zheng Xuefeng2 Tu Xuyan2
1. Henan Polytechnic University, school of computer science and technology, Henan, China
2. University of Science and Technology Beijing, school of information and engineering, Beijing, China
Email: Sjq0273@sina.com

Abstract

Based on the study of principle and approaches of generalized artificial life, humanoid grid management model (*HGMM*) is proposed. Refer to the characters and functions of human body control management system and successful artificial society model, the management scheme and strategy of *HGMM* are discussed. *HGMM* has humanoid management characteristics such as dual management scheme, centralized-decentralized management pattern and multi-level coordination function. This paper also proposes a cognitive model to enhance the collaboration among all participators in grid. At the end of this paper, an agent-oriented frame of grid is implemented, and a medium service model and a rational negotiation mechanism are also given.

Keyword: multi-agent generalized artificial life, humanoid grid management model.

1. Introduction

Today is an information epoch; the ways in which people produce, obtain and use information are multiform. But one of the key issues is how to make people quickly to get concise and integrated information. Grid is a new technology that may be helpful in information integration and sharing, and quick access and gathering of information. Grid mainly provides no different resource share for the high level applications. Its main research aspects include enrolling, expression, location, management, storage and accessing mechanism of all kinds of grid resource, and developing a whole intelligent information processing platform on which the grid user can conveniently issue, process and gain information.

Artificial life (*AL*) is the simulation, extension, expansion of nature life, as well as the improvement and extension of artificial intelligence. Generalized artificial life (*GAL*) is a man-made system of nature life in terms of the performance and behavior [3]. This paper proposes the humanoid grid management model (*HGMM*) based on *GAL*, and presents the dual management scheme, centralized-decentralized management pattern, and multi-level coordination function in *HGMM*. The objectives of the research on *HGMM* are to improve the intelligent and automatic information processing, realize information auto-collaboration and sharing in the grid, and improve the quality of grid service.

2. Grid and OGSA

Grid is a new technology built on Internet. It breaks through limits imposed on the compute resource, and makes people use grid resource by a brand-new, more freedom and more convenient mode, and solve more mixed problems.

Open Grid Service Architecture (*OGSA*) is a popular architecture of grid. It describes a set of standard protocols and interfaces to enable interoperability between different virtual organizations (*VO*) and their proprietary grid solutions. *OGSA* gives support to grid services that encapsulate any resource on the grid. This is achieved by integration of Computing Grid and Web Services within a comprehensive framework. Upon this, standard grid functionalities for creation, registry, discovery, and lifecycle management as well as secure and reliable invocation are defined. Thus each resource is represented as a Grid Service; therefore resource discovery mainly deals with the problem of locating and querying information about useful Grid Services. The reference implementation of *OGSA*, the Globus Toolkit 4 (*GT4*), supports resource discovery by introducing a hierarchical information service.

3. Humanoid Grid Management Model (*HGMM*)

Grid mainly uses the existing Internet/Intranet basic establishment, protocol, criterion, WEB technology and DB technology to provide an intelligent information processing platform for grid user. On the platform, information processing is distributed, collaborative and intelligent, and any grid user can visit all information in any place and at any time through a single register entrance [4, 5, and 6]. The final object of grid will be providing service on demand. So constructing a *HGMM* is necessary.

HGMM, based on generalized artificial life, can be expressed by formula (1).

$$\text{GAL} + \text{GMM} \rightarrow \text{HGMM} \qquad (1)$$

In “formula (1)”:

GAL --- Generalized Artificial Life.
GMM --- Grid Management Model.
HGMM --- Humanoid Grid Management Model.

On the basis of the research and theory of the “Generalized Artificial Life”, “Large system Cybernetics” and grid computing, we propose *HGMM* model on the basis of following ideas [1, 2, 3] .

1) Dual management scheme: There are dual management scheme of nerve dividing-area management and hormone dividing-work management in the human body management system. And there are dual management schemes in the society, for example, “two systems in a country” in china, nation administration management scheme ---- “strip and block integrating”. To realize reasonable, available and optimal integration of the dividing-area management and dividing-work management, we can apply the models and methods of generalized artificial life (manual nerve system, manual incretion hormone system, manual society model etc.) to study the dual management scheme of grid.

2) Centralized-decentralized management pattern: Whether human body’s control management system, human society management system or engineering technology

The research is supported by the National Natural Science Foundation of China (No.60375038)

domain, they are centralized-decentralized management pattern. For example, centre nerve control and periphery nerve control in human body, "macrocosmic control and microcosmic free" in my country's economy management pattern, distributed control system (*DCS*) etc.

3) Multi-level coordination function: The key of centralized-decentralized management is coordinate, and the stand and fall of centralized-decentralized management is also coordinate. There are many multi-level cooperative mechanisms in human body control management system, and these are worth to learn and refer for us. For example, coordination control between nerve and hormone, nerve quick speed control and hormone slow speed control, nerve dividing-area control and hormone dividing-work control, and so on.

4) Cognitive model: In our every day life, the basis of collaboration among people is man's cognizance. Cognizance is the precondition and basis of carrying on efficient cooperation among people, and it is awareness of many problems, such as: what is the work object? What is the task? Where should I gain information from? What information should I gain? Who should I interchange with? etc.

3.1 Dual Management Scheme in *HGMM*

HGMM based on *GAL* should have humanoid dual management scheme – "nerve dividing-area" management and "hormone dividing-work" management.

--First, "nerve dividing-area" management system.

The nerve system of human body has the dividing-area management function. For example:

* There are corresponding dividing-area projecting relations between the high-level nerve centre and every part of the body.

* The low level nerve centre – spinal cord has 31 nerve section, and differently manage every part, e.g. head, neck, chest, waist, trail, upper limbs, lower limbs.

* Peripheral nerve system include sense nerve and motor nerve distribute every part of the body, can receive outside stimulating, input the location info, output the location control info, and manage every part of the body.

--Second, "hormone dividing-work" management system.

The hormone system of human body has the dividing-work management function. Such as:

* Thyroxin can adjust and control the metastasis.

* Adrenaline can adjust and control the blood pressure.

* Insulin can adjust and control blood sugar.

Because different hormone take different cryptogram, and relevant target cell can only unscramble the cryptogram and accept the info, so hormone can control and manage all human body's apparatus and physiology function, and realize dividing-work management.

How to realize the "dividing-area" and "dividing-work" management in grid? We can apply the models and methods of artificial life to study *HGMM*. The information service center of grid is a set of servers, and every server has specific dividing-work. According to the content and attribute of the providing service, the service provider is divided to a virtual organization that can fulfill some specified services, so the dividing-area function is achieved.

3.2 Centralized - Decentralized Management Pattern in *HGMM*

The research of "Large System Cybernetics" indicates which centralized-decentralized management pattern is universal pattern in various kinds of large system. Such as:

* Human body control and management system: There are centrum nerve system and periphery nerve system in the nerve system. There are hormone centre, e.g. pituitary, and all kinds of endocrine hormone system, e.g. hypothyroid, pancreas. They are centralized-decentralized management pattern.

* Large society system: The organizations and groups in human society universally adopt the centralized-decentralized management and control. Such as example, army system, party system, nation system.

* Large engineering system: Centralized-decentralized management pattern is universal in the engineering technology domain, such as distributed control system (*DCS*), multi-level computer control management system etc.

In *HGMM*, centralized-decentralized management pattern should be adopted. But the key is how to integrate, unite and coordinate between centralized management and decentralized management. We can learn, simulate and summarize from human body control system and succeed human society party. The information service centre can be designed a relax centre, only takes charge the function such as registration, service location, service binding etc. Once gaining the authorization of the information service centre, the binding is erected between the service provider and the service requester. And all transactions can be coordinately completed between them under no interfering of the information service centre.

3.3 Multi-Level Coordinative Function in *HGMM*

The key of centralized-decentralized management is coordination. There are many multi-level coordinative mechanisms in human body control system. And we can learn and use for reference. For example, there are multi-level coordinative functions in nerve system, such as:

* Pallium's reflecting coordination.

*Halamencephalon's sense info fusion and coordination.

* Cerebellum's sport coordination and gesture balance.

* Spinal cord's segment coordination.

* Dual direction coordination between interaction and vice interaction.

There are multi-level coordinative functions in hormone system, such as:

* Coordination between hormone centre such as pituitary and all kinds of endocrine such as hypothyroid, pancreas.

* Dual direction coordination among incretion and hormone.

In the grid, we should study negotiation, collaboration and coordination among information service centre, service provider and service requester. The collaboration is realized by coordination and negotiation. Because information service centre, service provider and service requester is all self-governed entity and virtual artificial life, we can learn human body control system and human society system to develop many methods and technology about coordination, negotiation and collaboration. Such as: primary and secondary coordination, circulating coordination, grouping coordination etc.

3.4 Cognitive Model

In *HGMM*, we can erect a cognitive model (*CM*) to help

information service centre, service provider or service requester to charge the environment, gain correlate knowledge and project and organize itself behavior, thereby realize the collaboration among them. The *CM* is shown as formula 2:

CM = {Context, Constraint, History, Goal, Providing, Bias, GM, BCA, PCI} **(2)**

In "formula (2)":

Context --- describing the status of cognitive system in a moment.

Constraint --- the rule of solving problem.

History --- the acting series which have had happened.

Goal --- the final goal.

Providing --- the acting set which the cognitive system will plan to implement.

Bias --- the orientation's acting selection based on the environment, knowledge, and percept and cognitive.

GM --- Goal Matching.

BCA --- Bias Construction and Assessment.

PCI --- Plan Construction and Implementation.

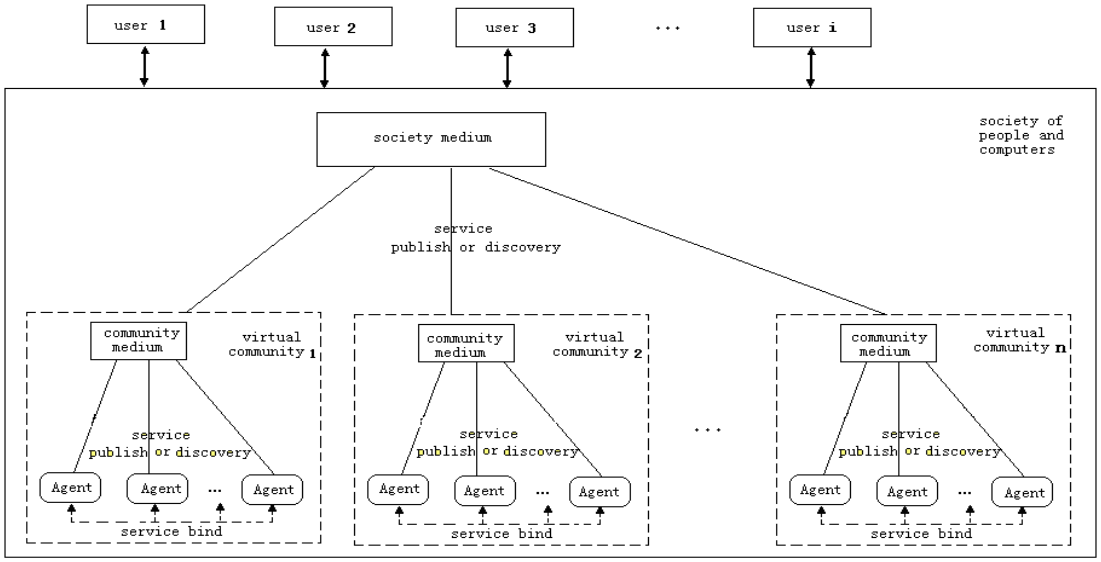


Fig. 1. The implementation frame of grid

4. Agent-oriented Implementation

Grid is a distributed, autonomic, dynamic system. Based on the functions and characters of *HGMM*, a frame of grid has been constructed (as shown in figure 1). It has dual management scheme, multi-level coordinating function, centralized-decentralized management pattern [7, 8].

4.1 Implementation Frame

The implementation frame of grid has an agent-oriented and fractal structure. Society of people and computers is composed of virtual communities, and virtual community is composed of agents. Any two or several agents can unite through Internet or Intranet. So the frame is flexible, extendable and reduced.

Society of people and computers is an open society of people and computers. In summary, Society of people and computers has 5 characters: people and computers, open society, sophisticated interaction, 3A using (any place, any time, and any device) and high quality of service. Society of People and Computers (*SPC*) is the set of Virtual Community (*VC*):

SPC = {VCi}

VC*i*= {a:Agents | ∋ u: Users and (a, u) ∈ Binding*i* } **(3)**

In "formula (3)":

VC*i* — No. i Virtual Community.

Agents — the set of agent.

Users — the set of grid user.

Binding*i* — the binding relations between agent which have registered No. i virtual community and relevant grid user.

Virtual community is a virtual organization. It can provide special function and service, such as tour service, weather forecasting, finance service, education service etc. By the management of virtual community, hypo-structure collaboration mechanism is erected and maintained, and the collaboration among agents on the administrative levels can be promoted.

In the frame, society medium and community medium play the role of service medium, and agent plays the role of the service provider or the service requester. There are a relaxed, managing and managed relation among society medium, community medium and agent.

4.2 Medium Service

Service medium charges the coordination management task such as service register, resource location, service binding etc. Agent provides some grid services on demand, dispatches itself resource, and manages itself transaction. But agent must exchange information with service medium so that service medium timely grasps their resource information, dynamically alter the management decision-making based on the actual circs, and coordinately settles such problems as "contending use", "monopolize" etc. In addition, Agent can only use the grid resource behind authorization of service medium. Once registering successfully and gaining authorization from the service medium, agent independently and freely works without the interference of the service medium.

The medium service (*MS*) provided by "society medium" and "community medium" can be expressed by formula (4):

MS= {GS, SM, GSLD, BKD, MSM, SCA} **(4)**

In "formula (4)":

GS -- Grid Service. *GS* includes such basic service description as service name, text describing etc, such service function description as service function, service behavior etc, such service non-function attribute description as cost, response time, priority etc.

SM -- Service Medium. *SM* is made up of society medium and community medium. *SM* is in charge of gathering the published grid service (*PGS*) and storing the services into *GSLD* according to the type of service. When service requester needs gaining service, firstly it must send a requesting grid service (*RGS*) to *SM*, behind receiving the *RGS*, *SM* starts running *SCA* and compose the service of RGS describing from *GSLD* and *DKD*.

GSLD -- Grid Service Local Database. For "society medium", *GSLD* stores the whole restricting relations of all activities, such as role mark, role ability, location of community medium, and relations with other community medium. For "community medium", *GSLD* stores the local restricting relations of the activities of some special domain, such as role mark, role ability, location of agent, and the restrictions with other agent interaction.

BKD -- Background Knowledge Database. *BKD* stores such knowledge as general knowledge of a domain, tested and verified knowledge of domain experts and composing or evaluating model of domain experts etc, Background knowledge decide the depending relations among services, and it can be used as assistant decision-making in *SCA*.

MSM -- Medium Service Mechanism. *MSM* is event-driven. Medium service will be driven by such events as registering, publishing, discovering etc.

SCA -- Service Composing Algorithm. *SCA* is the key that decides high or low capability of medium service, and it is emphasis that we are going to study. At present, there are such many correlative algorithms as kinds of UDDI system based on syntax, augment UDDI Registry based on semantics etc. From the whole, the algorithm based on syntax can be easily implemented, but low precision, not enough stand for service composing and validating; the algorithm based on semantics is high precision, but too complex, non-flexible and poor practicability [9, 10, 11].

4.3 Rational Negotiation

Agent is an autonomous entity. And it has many characters, such as autonomy, activity, reactivity, mobility, intelligence etc. Agent manages itself resource, and provides services for another agent according to the as-needed pattern. Agent is capable of use knowledge to alter itself thinker state so that adapting the environment change and cooperatively settling problem. Once a joint is erected between agents, and no need the interference of service medium, all transaction can be completed by coordination and negotiation. The cooperation and negotiation can be implemented by constructing a mind model (*MM*). The mind model can help agent to make a rational decision that the agent should adopt negotiating action and content [12]. *MM* can be expressed by 4 terms:

$$MM = \{SB, BD, NS, CK\} \quad \textbf{(5)}$$

In "formula (5)":

SB -- sociality belief. Including behavior inhibition conditions and information of current sociality environment.

BD -- base desire. *BD* describes the actions that the agent is going on to do, the requirements and action processes etc.

NS -- negotiation status. *NS* records the dynamic information of negotiating process.

CK -- consequence knowledge of negotiating behavior and content. Aiming to apply *SB*, *BD*, *NS* to decide the negotiating situation of the agent should adopt and the content of the cooperative contract. The consequence knowledge can be divided evaluating knowledge, stratagem knowledge and tactics knowledge.

5. Conclusions

The paper had proposed *HGMM* based on the theory and method of generalized artificial life. And this paper also had discussed the designing idea about dual management scheme, centralized-decentralized management pattern and multi-level coordinative function. And an agent-oriented frame of grid was designed. On the basis of these, we can go on study the concrete implementing technology and algorithm.

6. References

[1] X.Y.Tu, X.S.Li et al, Intelligent Management. Tsinghua University Press, 1995

[2] X.Y.Tu, Z.Wang et al, Large System Cybernetics. Beijing University of Posts and Telecommunications Press, 2005

[3] X.Y.Tu, "General Artificial Life", Journal of University of Science and Technology Beijing (Artificial Life special) , 2002

[4] Z.W.Xu, B.M.Feng, et al, Grid Computing Technology [M], Publishing House of Electronics Industry, 2004.5

[5] H.Jini, P.P.Yuan, et al. The Grid 2[M], Publishing House of Electronics Industry, 2004.10

[6] MCQuay, WK, The collaboration grid: Trend for next generation distributed collaborative environments, Enabling Technologies for simulation Science. 2004: 359-366

[7] I. Foster, C. Kesselman, et al, Grid Services for Distributed Systems Integration, IEEE Computer, 2002,35 (6), 37-46

[8] Z.W.Xu, X.L.Li, et al, Architecture study of VEGA information grid, Journal of computer research and development. 2002,39(8), 948-951

[9] G. Weiss, "Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence", the MIT Press: Cambridge, Massachusetts, 1999.

[10] H.J. Mueller, Towards Agent Systems Engineering, Data & Knowledge Engineering, 1997, 23(3), 217-245.

[11] J.Gao, C.H.Yuan, et al, SASA5: A Method System for Supporting Agent Social Activities, Chinese Journal of Computer, 2005, 28(5), 838-848.

[12] C. Langton(Ed), Artificial Life, Volume VI of SFI Studies in the Sciences of Complexity, pages xiii-xviii and pages1-47, Addison-Wesley, Redwood City, CA, 1989.

Design of a Cultural Algorithm based on an Improved Evolutionary Programming to Solve Constrained Optimization Problems *

Sheng Liu, Xingyu Wang, Xiaoming You
College of Information Science and Engineering, East China University of Science and Technology
Shanghai 200237, China
Email: ls6601@163.com
Or
Sheng Liu
School of Management, Shanghai University of Engineering Science
Shanghai 200065, China

ABSTRACT

A cultural algorithm is analyzed and designed. The application of this algorithm for solving complex constrained optimization problems has also been discussed. In this approach, an improved evolutionary programming is used as a population space, in which a shift factor is proposed; according to the corresponding population space, the knowledge sources contained in the belief space of cultural algorithm are specifically designed and are used to guide the evolutionary search. The approach not only can maintain quite nicely the population diversity, but also can help to converge to the global optimal solution rapidly, which is validated by some experimental results.

Keywords: Constrained Optimization, Evolutionary Computation, Cultural Algorithm, Evolutionary Programming.

1. INTRODUCTION

The cultural algorithm (CA) is an evolutionary computational model derived from observing the cultural evolution process in nature, which is made of two main components: the population space and the belief space [1,2]. First, individuals in the population space are evaluated with a performance function obj(). Then, an acceptance function accept() will determine which individuals are to impact the belief space. Experiences of those chosen elites will be used to update the knowledge/beliefs of the belief space via function update(), which represents the evolution of beliefs. Next, the beliefs are used to influence the evolution of the population via function influence(). New individuals are generated under the influence of the beliefs, and from then, together with old individuals, individuals are selected and form a new generation of population. The population component and the belief space interact with and support each other. Those interactions are depicted in Fig. 1.

In this paper, we propose the use of a cultural algorithm as a global optimization technique, in which an improved Evolutionary Programming (EP) is selected to model the population space component; according to the characteristics of EP, the belief space is divided in four knowledge sources, the different influence factors is analyzed and designed in detail. By using them, the influence of each knowledge source played in the evolutionary process can be dynamically adjusted, which will lead to an efficient optimization process.

* The National Natural Science Foundation of China under Grant No. 60543005 supports this work.

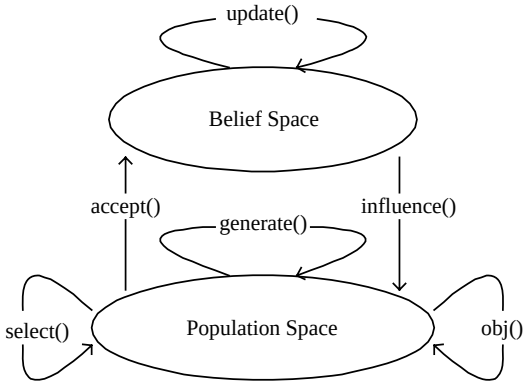


Fig. 1. A framework of cultural algorithms

2. EVOLUTION OF THE POPULATION SPACE

When cultural algorithms were applied to real parameter optimization, there were three dominant population-only evolutionary algorithms (GA, ES, and EP) can be used to model the population space. In our study, the EP was selected to model the population space component, because it is a model used frequently in real-valued function optimization with better performance [3,8]. The CEP (Classical Evolutionary Programming) model and its main algorithmic steps are discussed as follows:

1) Generate the initial population of p individuals at random, where p is the number of individuals.

2) Compute the performance score for each individual, using the given current objective function—obj().

3) Generate p new offspring solutions from the current p parents with a single parent for each child. Note that the total number of individuals now is $2p$.

4) Compute the performance score for each new offspring using the current performance.

5) Apply tournament selection on the combined population of $2p$ individuals, and compute the number of wins for each individual according to the following:

$$\text{for} \quad i=1 \quad \text{to} \quad c$$

$$wins = wins + \begin{cases} 1 & if \ \ f(x) \le f_i(x) \\ 0 & otherwise \end{cases}$$

where c is the number of competitors randomly selected from the population, and $f_i(x)$ is the competitor fitness value.

6) Select the p individuals that have the greatest number of *wins* to be the parent for the next generation.

7) While exit conditions are not satisfied, go back to step3.

The tournament selection algorithm described above is the selection method frequently used in EP [4], although the method guarantees the survival of the best individual, since it will always win and be given a guaranteed maximum number of wins. It is possible, however, to ensure survivability of the best individual by breaking the ties between individuals having the same number of wins by sorting them "locally" relative to their fitness values. So, this method has merely measured superiority order between the individuals, has not measured shift size between the current individual and the previous, it is easy to produce many similar optima, but difficult to produce distributed broadly optima.

In order to maintain diversity, an individual's shift size must take into account when calculating the fitness function. So, the shift factor is proposed in this paper and is computed in step4) using the following function:

$$dv_i = dv_i + \left| x_i^{(current)} - x_i^{(previous)} \right|$$

where $x_i^{(current)}$ and $x_i^{(previous)}$ are the values of *i*-th decision variable respectively corresponding to the best performance score found so far and the previous.

When individuals having the same number of wins by sorting them "locally" relative to their fitness values, the individuals who has greater shift factor will be given priority to be selected. It is a very important technique that allows EP to adapt a new path quickly when selected individuals have the same number of wins.

3. DESIGN AND INFLUENCE OF THE BELIEF SPACE

The second space is the belief space, which provides an explicit mechanism for acquisition, storage and integration of individual and group's problem solving experience and behavior. In our approach, the belief space is divided in four knowledge sources [5,6], according to the characteristics of EP, the different influence factors is analyzed and designed as follows.

1) Situational Knowledge: Situational knowledge consists of the best exemplar E found along the evolutionary process. It represents a leader for the other individuals to follow. The variation operators of the population space evolution are influenced in the following way:

$$x'_{i,j} = \begin{cases} x_{i,j} + \left| (x_{i,j} - E_j) * N_{i,j}(0,1) \right| & \text{if } x_{i,j} < E_j \\ x_{i,j} - \left| (x_{i,j} - E_j) * N_{i,j}(0,1) \right| & \text{if } x_{i,j} > E_j \end{cases}$$

where $x_{i,j}$ is the *j*-th parameter of the *i*-th individual, $x'_{i,j}$ is the offspring of $x_{i,j}$, E_j is the *j*-th component of the individual *E*, $N_{i,j}(0,1)$ is a normally distributed variable with an expectation of 0 and variance 1. The update of the situational knowledge is done by replacing the stored individual, *E*, by the best individual found in the current population, x_{best}, only if x_{best} is better than E.

2) Normative Knowledge: The normative knowledge contains the intervals for the decision variables where good solutions have been found, in order to move new solutions towards those intervals. Thus, the normative knowledge has the structure shown in Fig. 2.

l_1	u_1	l_2	u_2	…	l_n	u_n	dm_1	dm_2	…	dm_n
L_1	U_1	L_2	U_2	…	L_n	U_n				

Fig. 2. Structure of the normative knowledge

In Fig. 2, l_j and u_j are the lower and upper bounds, respectively, for the *j*-th decision variable; L_j and U_j are the values of the fitness function associated with that bound. Also, the normative knowledge includes a scaling factor, dm_j, to influence the mutation operator adopted in the population space evolution. The following expression shows the influence of the normative knowledge on the variation operators:

$$x'_{i,j} = \begin{cases} x_{i,j} + \left| (u_j - l_j) * N_{i,j}(0,1) \right| & \text{if } x_{i,j} < l_j \\ x_{i,j} - \left| (u_j - l_j) * N_{i,j}(0,1) \right| & \text{if } x_{i,j} > u_j \\ x_{i,j} + \dfrac{u_j - l_j}{dm_j} * \left| (u_j - l_j) \right| * N_{i,j}(-1,1) & \text{otherwise} \end{cases}$$

The update of the normative knowledge can reduce or expand the intervals stored on it. An expansion takes place when the accepted individuals do not fit in the current interval, while a reduction occurs when all the accepted individuals lie inside the current interval. The values dm_j are updated with the greatest difference $\left| u_j - l_j \right|$ found during application of the variation operators of the previous generation.

3) Topographical Knowledge: The usefulness of the topographical knowledge is to create a map of the fitness landscape of the problem during the evolutionary process. It consists of a set of cells, and the best individual found on each cell. The topographical knowledge, also, has an ordered list of the best cells, based on the fitness value of the best individual on each of them. For the sake of a more efficient memory management, we use a spatial data structure, called *k*-d tree see Fig. 3. In *k*-d trees, each node can only have two children. Data structure of each node (see Fig. 4) contains lower and upper bounds for *n* decision variables indicating the ranges associated with the best solutions found so far, and a pointer to its children.

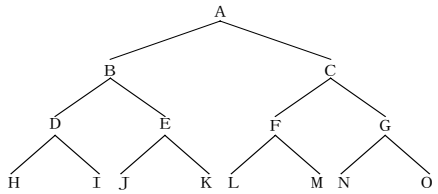


Fig. 3. Data structure of the topographical knowledge

Interval lower limit(l_1, l_2, … l_n)
Interval upper limit(u_1, u_2,…u_n)
Best solution(x_1, x_2, …x_n:f)
Pointer to the children: null

Fig. 4. Data structure of each node

The influence function tries to move the children to any of the best cells in the list:

$$x'_{i,j} = \begin{cases} x_{i,j} + \left|(u_{c,j} - l_{c,j}) * N_{i,j}(0,1)\right| & if \ x_{i,j} < l_{c,j} \\ x_{i,j} - \left|(u_{c,j} - l_{c,j}) * N_{i,j}(0,1)\right| & if \ x_{i,j} > l_{c,j} \\ x_{i,j} + (u_{c,j} - l_{c,j}) * N_{i,j}(0,1) & \text{otherwise} \end{cases}$$

where $l_{c,j}$ and $u_{c,j}$ are the lower and upper bounds of the cell *c*, randomly chosen from the list of the best cells. The update function splits a node if a better solution is found in that cell, and if the tree has not reached its maximum depth. The dimension in which the division is done, is the one that has a greater difference between the solution stored and the new reference solution.

4) History Knowledge: The history knowledge contains information about sequences of environmental changes in terms of shifts in the distance and direction of the optimum in the search space. History knowledge records in a list, where the location of the best individual is found before each environmental change. That list has a maximum size *w*. The structure of history knowledge is shown in Fig. 5, where e_i is the best individual found before the *i*-th environmental change, ds_j is the average distance of the changes for parameter *j*, and dr_j is the average direction if there are changes for parameter *j*. In our study, if we are trapped in a local optimum, history knowledge can be used to adjust shift size of distance and direction.

e_1	…	e_i	…	e_w

ds_1	ds_2	…	ds_n
dr_1	dr_2	…	dr_n

Fig. 5. Structure of the history knowledge

The expression of the influence function of the history knowledge is the following:

$$x'_{i,j} = \begin{cases} e_i.x_j + dr_j * random(0,1) & \alpha\% \\ e_i.x_j + ds_j * random(-1.5,1.5) & \beta\% \\ random(l_j, u_j) & \varphi\% \end{cases}$$

$$ds_j = \frac{\sum_{k=1}^{w-1} \left| e_{k+1}.x_j - e_k.x_j \right|}{w-1}$$

$$dr_j = \mathrm{sgn}\left(\sum_{k=1}^{w-1} \mathrm{sgn}(e_{k+1}.x_j - e_k.x_j)\right)$$

where $x'_{i,j}$ is parameter *j* for the newly generated individual i, $e_i.x_j$ is parameter *j* of the best solution from the last change event, l_j and u_j are the lower and upper bounds, respectively, for the *j*-th decision variable, the function *sgn(a)* returns the sign of *a*.

The shift manner for newly generated individual is decided by using a roulette wheel approach. The roulette wheel selects the area in which to generate individuals relative to the moving direction α %, moving distance β %, and the entire domain range φ %.

4. SIMULATION EXPERIMENT AND RESULTS ANALYSIS

To validate our approach, we adopted the well-known benchmark nonlinear constrained problem included in [7]. This problem features a convex objective function subject to a nonlinear equality constraint, which is also a "ridge" problem that is generally difficult for evolutionary algorithm. The problem is given below:

$$\min Z = -12x - 7y + y^2$$

Subject to: $0 \le x \le 2,\ 0 \le y \le 3,\ y \le -2x^4 + 2$

The parameters used by our approach are the following: The original population *size* is 50, maximum number of generations =200, maximum depth of the *k*-d tree = 12, the size of the list in the history knowledge *w*=5, $\alpha = \beta = 0.45$, φ =0.1 and the length of the best cells list = 10. Each program was run 100 times for this problem. Each run was stopped after 200 generations. The statistical results compared to the results obtained by the CEP method and the results included in [7] are shown in table 1.

Table1. Comparison of the results for the test function

Algorithms (each runs 100times)	Mean of Generations	Standard Deviation of Generations
Classical EP	>40	>30
Hierarchical Architecture model	>8.61	>2.62
The Results of Our Approach	>7.5	>2.14

As shown in table1, it takes only7.5 generations on the average for our approach and it is very robust as suggested by the relatively small standard deviation in the number of generations taken. The main reason is that the constraints knowledge stored in belief space is used to guide the search of the evolutionary algorithm very efficiently, avoiding that it moves to unpromising regions of the search space.

5. CONCLUSIONS

We have presented an approach based on a cultural algorithm and an improved evolutionary programming for constrained optimization. Adding a belief space to an improved evolutionary programming algorithm, the approach has provided highly competitive results at a relatively low computational cost. This suggests that the proper use of domain knowledge may certainly improve the performance of an evolutionary algorithm. Also, it suggests that such domain knowledge may be extracted during the evolutionary process in which we aim to reach the global optimum of a problem.

6. REFERENCES

[1] R.G.Reynolds, "An Introduction to Cultural Algorithms", In: Proceedings of the 3rd annual Conference on Evolution Programming, River Edge, NJ: World Scientific Publish, 1994,131-139.

[2] C.J.Chung, R.G.Reynolds, "A Testbed for Solving Optimization Problems using Cultural Algorithms", In Proceedings of the Fifth Annual Conference on

Evolutionary Programming, Cambridge, Massachusetts: MIT Press, 1996,225-236.
[3] D.B.Fogel, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, Piscataway, NJ:IEEE Press, 1995.
[4] T.Back, G.Rudolph, "Evolutionary Programming and Evolution Strategies: Similarities and Differences", In Proceedings of 2nd Annual Conference on Evolutionary Programming, San Diego, CA, 1993,11-22.
[5] S.M. Saleem, " Knowledge-Based Solutions to Dynamic Problems Using Cultural Algorithms", Detroit Michigan: Wayne State University, Ph.D thesis, 2001.
[6] R.L. Becerra, C.A. Coello Coello, "Culturizing Differential Evolution for Constrained Optimization," In: ENC'2004, Mexico:IEEE Service Center, 2004,304-311.
[7] X.D. Jin, "Solving Constrained Optimization Problems Using Cultural Algorithms and Regional Schemata", Detroit Michigan: Wayne State University, Ph.D thesis, 2001.
[8] H.Schwefel, Numerical Optimization of Computer Models, John wiley, 1981.

Nonlinear Optimization Without Restriction Based On Quantum Genetic Algorithm

Bei Huang, Shitong Wang
School of Information Engineering, Southern Yangtze University
Wuxi, Jiangsu 214122, China
Email: RainBella6@163.com, wangst@yahoo.com.cn

ABSTRACT

Quantum Genetic Algorithm (QGA), which is based on Quantum Computation and Genetic Algorithm, has better searching capability and quicker convergence speed since it introduces qubit and quantum rotation gate into GA. Nonlinear optimization without restriction is a typical engineering application. However, conventional optimization methods are time-consuming and can be easily trapped in the local optimum. In this paper, an experiment is carried out on a typical complex nonlinear optimization problem using QGA. The result of the experiment shows that Quantum Genetic Algorithm has powerful searching ability and that it is practical and efficient in this field.

Keywords: Genetic Algorithm, Quantum Genetic Algorithm, Qubit, Quantum Rotation Gate, And Nonlinear Optimization.

1. INTRODUCTION

Quantum Genetic Algorithm (QGA) is based on Quantum Computing and Genetic Algorithm (GA), which has better searching ability and quicker convergence speed.

In the early 1980s, Benioff and Feynman proposed the concept of quantum computing [1, 2]. The quantum computing uses the superposition, entanglement and coherence characters of quantum states so that it is likely to resolve the NP problem in classic computing. Since then, the quantum computing has attracted wide attention and soon become the hot topic of research, especially after Shor's quantum prime factoring algorithm [3] and Grover's random database search algorithm [4] were proposed.

In 2000, Han proposed the Quantum Genetic Algorithm (QGA). The algorithm used the quantum state vector description and the quantum rotation gate operation. Its main contribution is to apply qubit's probabilistic amplitude representation to the coding of the chromosome. However, this algorithm is mainly used to resolve the 0-1-knapsack problem [5, 6].

Optimization of complex nonlinear function is a typical problem in engineering application. It includes nonlinear optimization with restriction and nonlinear optimization without restriction. Nonlinear optimization without restriction has a wide application in science and industry and it is the foundation of the further research of restrictive optimization. However, conventional solution of complex nonlinear optimization is usually not satisfying. For example, it may be stuck at a local optimum.

In QGA, the adoption of qubit coding enables one chromosome to represent the superposition of multi-states simultaneously and makes the QGA superior in diversity to the classic genetic algorithm. And since it introduces quantum rotation gate into GA, QGA has better search ability and quicker convergence speed. The experiment shows that Quantum Genetic Algorithm is practical and efficient in complex nonlinear optimization without restriction.

2. QUANTUM GENETIC ALGORITHM

The QGA is based on the representation of the quantum state vector. It applies the probabilistic amplitude representation of qubit to the coding of chromosome, uses quantum logical gates to realize the update operation, and eventually reaches the optimum resolution of the goal [7].

2.1 Qubit Encode

In Conventional Genetic Algorithm (CGA), chromosome is represented through a definite value, such as classic bit, while in Quantum Genetic Algorithm (QGA), chromosome is represented by qubit. The difference between the qubit and the classic bit is that qubit can be in basic state '1' or basic state '0', or any superposition of the two quantum states simultaneously, which means the state '1' and the state '0' exist at the same time with a certain probability respectively and appear state '1' or state '0' in the measuring procedure.

For easy writing and calculation, the basic quantum states can be denoted as $|0\rangle$ and $|1\rangle$, and a superposition of the two quantum states can be expressed as a unit vector[8] $|\psi\rangle$ in the two-dimensional Hilbert space, as illustrated in Fig. 1..

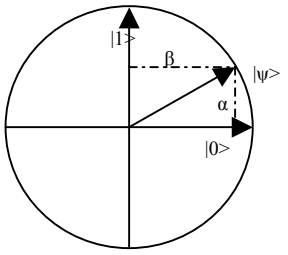


Fig. 1. Superposition of Two Quantum States

In this space, $|\psi\rangle$ can be represented as Eq. (1):

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

where the complex number α and β represent the probabilistic amplitude of corresponding state and satisfy the condition $|\alpha|^2 + |\beta|^2 = 1$, in which $|\alpha|^2$ is the probability of $|0\rangle$ and $|\beta|^2$ is the probability of $|1\rangle$.

The probabilistic amplitude of a chromosome with *m* genes

can be expressed as:

$$q_j^t = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1k} & \alpha_{21} & \cdots & \alpha_{2k} & \cdots & \alpha_{m1} & \cdots & \alpha_{mk} \\ \beta_{11} & \cdots & \beta_{1k} & \beta_{21} & \cdots & \beta_{2k} & \cdots & \beta_{m1} & \cdots & \beta_{mk} \end{bmatrix} \quad (2)$$

Where q_j^t represents the chromosome of the *j-th* individual in the *t-th* generation, *k* is the qubit number of every gene, *m* is the gene number in each chromosome.

In QGA, qubit is used to store and represent one gene which may be in the '1' state, the '0' state, or any superposition of the two. Thus the information represented by this gene is not stable, but probable; therefore, when an operation is carried out on this gene, it may be done to all probable information simultaneously. The adoption of qubit makes the QGA superior in diversity to the classic genetic algorithm. For the same optimization problem, the population size of QGA can be much less than that of the Conventional Genetic Algorithm (CGA). Convergence can be also obtained with the qubit representation. As $|\alpha|^2$ or $|\beta|^2$ approaches to 0 or 1, the qubit chromosome converges to one single state.

2.2 The Structure of QGA

The structure of QGA is described as follows:

1) initialize the population $P(t_0)$;

2) make $R(t_0)$ through measuring $P(t_0)$ states;

3) evaluate $R(t_0)$;

4) store the best individual among $R(t_0)$ and its fitness;

5) while (not termination condition) do
   begin
   {
   t=*t*+1;
   make $R(t)$ through measuring $P(t)$ states;
   evaluate $R(t)$;
   update $P(t)$ by quantum logical gates $U(t)$
   so as to get next generation $P(t+1)$
   get next generation $P(t+1)$;
   store best individual and its fitness among $P(t)$;
   }
   end

The first step of QGA is to initialize the population with *n* individual $P(t_0) = \left\{ p_1^{t_0}, p_2^{t_0}, \cdots, p_n^{t_0} \right\}$, where $p_j^{t_0} \left(j = 1, 2, \cdots, n \right)$ is the *j*-th individual in the initial generation of population and $p_j^{t_0}$ can be represented as Eq. (2). All the genes $\left(\alpha_{ij}, \beta_{ij} \right)$ in the entire chromosome of the population are initialized to $\left(1/\sqrt{2}, 1/\sqrt{2} \right)$, which means that one chromosome may represent the superposition of all possible states with the same probability.

The second step is to measure all the individuals of the initial generation and thus obtains a group of definite solution $R\left(t_0\right) = \left\{ \gamma_1^{t_0}, \gamma_2^{t_0}, \cdots, \gamma_n^{t_0} \right\}$, where $\gamma_j^{t_0} \left(j = 1, 2, \cdots, n \right)$ is the *j*-th solution in the initial generation (the *j*-th individual measurement), represented as a length m binary string $\left(x_1, x_2, \cdots, x_m \right)$. Every bit $x_i \left(i = 1, 2, \cdots, m \right)$ is either 0 or 1, which is selected by the probabilistic amplitude of the qubit $|\alpha_{ij}|^2$ or $|\beta_{ij}|^2$, $\left(i = 1, 2, \cdots, m \right)$. The measuring procedure is to generate a random number γ between 0 and 1 and if $\gamma > |\alpha_{ij}|^2$, and then the measuring result is 1, otherwise 0.

The next step is to evaluate the group of solution with its fitness. The initial best individual and its fitness among the binary solution $R\left(t_0\right)$ are selected and stored as the aim of the later evolution.

In the "while" loop step, the solution of the generation is converged little by little to the optimum solution. In each iteration, a group of solutions $R\left(t\right)$ is obtained through measuring $P\left(t\right)$ and the fitness of every solution is calculated. Then according to the present evolutionary aim and the adjusting strategy set beforehand, the individuals of the generation are updated through applying the quantum logical gates to get $P\left(t+1\right)$, and the updated optimum solution is stored and compared with the present evolutionary aim. If the optimum solution is better than the evolutionary aim, the present evolutionary aim is replaced by the optimum solution; otherwise the present evolutionary aim remains unchanged.

2.3 Qubit Rotation Gate Strategies

Since Quantum Genetic Algorithm (QGA) updates probabilistic amplitude of quantum state by applying quantum logical gates to keep the diversity of the population, the updating method of quantum logical gates is the key point of QGA. In consideration of features of Genetic Algorithm (GA) and specific applications, Quantum rotation gate is more suitable for the updating operation of QGA [9]. Quantum rotation gate can be represented as:

$$G = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (3)$$

Where θ is the rotation angle, $\theta = k \times f\left(\alpha_i, \beta_i\right)$, in which k and $f(\alpha_i, \beta_i)$ are the value and the direction of the rotation angle respectively. The value of k has effect on the convergence speed; if the value is too big, the solution may diverge or have premature convergence to a local optimum. In this paper, the value of k can be adjusted between 0.1π and 0.005π according to the differences between the genetic generations. The function $f(\alpha_i, \beta_i)$ can make the probabilistic amplitude evolve toward the present evolutionary aim. The updating strategy is as Table. 1. shows:

Table 1. : Function $f(\alpha_i, \beta_i)$ Selection Strategy

		$f(\alpha_i, \beta_i)$	
$d_1 > 0$	$d_2 > 0$	$\mid \xi_1 \mid > \mid \xi_2 \mid$	$\mid \xi_2 \mid > \mid \xi_1 \mid$
True	True	+1	-1
True	False	+1	+1
False	True	-1	-1
False	False	-1	+1

In Table. 1., $d_i = \alpha_i \times \beta_i, \xi_i = \arctan\left(\alpha_i / \beta_i\right), i = 1, 2$, where (α_1, β_1) and (α_2, β_2) are the probabilistic amplitude of the present evolutionary aim and the current individual respectively. The updating strategy is described as follows: If $d_1 > 0$ and $d_2 > 0$, the current individual and the present evolutionary aim are at the same quadrant such as the first quadrant and the third quadrant. In this case, if $\mid \xi_1 \mid > \mid \xi_2 \mid$, the current individual should rotate anticlockwise toward the present evolutionary aim, i.e. $f(\alpha_i, \beta_i)$ should be +1, otherwise -1. The other three cases can be discussed in a similar way.

Thus, the third step in the structure of the Quantum Genetic Algorithm (QGA) can be described as updating $P(t)$ by applying quantum rotation gate $G(t)$. The updating operation is shown as follows:

$$p_j^{t+1} = G(t) * p_j^t \tag{4}$$

where superscript t is the evolving iteration, $G(t)$ is the quantum gate of the t-th generation, p_j^t is the probabilistic amplitude of a certain individual in the t-th generation, and p_j^{t+1} is the probabilistic amplitude of the corresponding individual in the t-th generation $(t' = t + 1)$. p_j^t and p_j^{t+1} can be represented as Eq. (2).

3. NONLINEAR OPTIMIZATION WITHOUT RESTRICTION BASED ON QGA

Optimization, the core subject in operations research and administrative science, is to search for the maximum or minimum of function with one or more variables under some constrained conditions. Nonlinear optimization includes nonlinear optimization with restriction and nonlinear optimization without restriction. Nonlinear optimization without restriction has a wide application in science and industry and it is the foundation of the further research of restrictive optimization.

Optimization of complex nonlinear function without restriction is a typical problem in engineering application. Conventional optimization methods are time-consuming and can be easily trapped in the local optimum. Being served as a brand-new effective optimization method, Quantum Genetic Algorithm (QGA) is coming under observation for its great potential. In this paper, the application of the nonlinear optimization based on QGA will be discussed.

3.1 Nonlinear Optimization Without Restriction and Ackley Function

Nonlinear optimization without restriction is to search for the maximum or minimum of a nonlinear function without any restrictive condition. Although most practical optimization problems must satisfy a certain restriction, nonrestrictive optimization is the foundation of the further research of restrictive optimization.

Generally, nonlinear optimization can be described as the following expression:

$$\min\left(f\left(x\right)\right), x \in \Omega. \tag{5}$$

Where f a nonlinear real-number function, feasible region is Ω is the subset of corpora E^n. And if $\Omega = E^n$, the problem is nonrestrictive completely.

Given a point $x^* \in \Omega$, if there is a value $\varepsilon > 0$, which makes any point $x \in \Omega$, whose distance from x^* is not greater than, ε satisfies $f(x) \geq f(x^*)$, then x^* is said to be the local optimum point in the region Ω. If all the points $x \in \Omega$ satisfy $f(x) \geq f(x^*)$, then x^* is said to be the global optimum point in the region Ω.

Among the problems of nonlinear optimization, one experimental function that can take the most advantage of the optimization method is Ackley function. Ackley function is a continuous function, which is modulated from exponential function, superposed with reasonably enlarged cosine wave and it can be represented as:

$$f\left(x_1, x_2\right) = -20 \times e^{-0.2 \times \sqrt{\frac{1}{2}\left(x_1^2 + x_2^2\right)}} - e^{\frac{1}{2}\left(\cos\left(2\pi x_1\right) + \cos\left(2\pi x_2\right)\right)} + 22.71282 \tag{6}$$

The figure of the function $f(x_1, x_2)$ in the interval $-5 \leq x_j \leq 5, j = 1, 2$ is demonstrated as Fig. 2. :

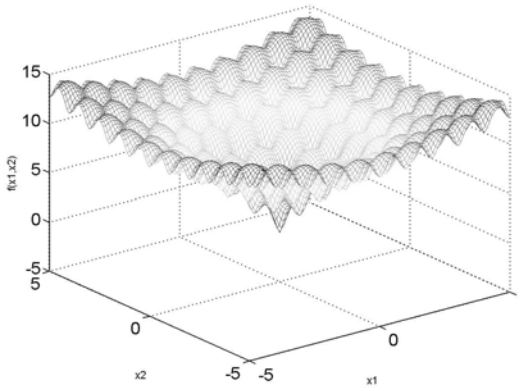


Fig. 2. the Figure of Ackley Function

From Fig. 2., it can be seen that the search for the optimum of Ackley function is quite complex since conventional methods can easily be trapped in the local optimum during the "climbing" procedure. In order to compensate this inadequacy, the search area should be extended so as to get across the disturbing valley and a better optimum point can be approached step by step. By introducing quantum state into the description of chromosome, QGA can obtain population diversity with fewer individuals and thus the search area of QGA can be

guaranteed. At the same time, with the updating method of quantum rotation gate, QGA is able to miss the disturbing valley and eventually reach the optimum resolution of the goal.

3.2 Experiment Results

The parameters of the QGA are set as follows: the size of the initial population is 10, the number of the qubits is 20, and the generation is 1000.

The result of the experiment is that the optimum found by QGA is $x_1 = 0.000000, x_2 = 0.000000$, and that the minimum of the Ackley function is $Ackley(x_1, x_2) = 0.005418$. In theory, the optimum is $x_1 = 0.000000, x_2 = 0.000000$, while the minimum of the Ackley function is 0 .

3.3 Conclusion

The experiment shows that complex nonlinear optimization without restriction can be resolved successfully with the powerful searching capability of QGA.

4. REFERENCES

[1] H. Tony, "Quantum computing: An introduction", Computing & Control Engineering, 1996, 10(3): 105-112.

[2] A. Narayanan, "An introductory tutorial to quantum computing", Proceedings of IEE Colloquium on Quantum Computing: Theory, Applications and Implications, London: IEE Press, 1997.

[3] P.W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", Goldwasser S,ed. Proceedings of the 35th Annual Symposium on the Foundation of Computer Sciences. Los Alamitos: IEEE Computer Society Press, 1994, 20-22.

[4] L.K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search", Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, ACM Press, 1996, 212-221.

[5] K.H. Han, J.H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problems", Proceedings of IEEE Conference on Evolutionary Computation, Piscataway: IEEE Press, 2000, 1354-1360.

[6] K.H. Han, K.H. Park et al, "Parallel quantum-inspired genetic algorithm for combinatorial optimization problems", Proceedings of the IEEE Conference on Evolutionary Computation, Piscataway: IEEE Press, 2001, 1442-1429.

[7] J.N. Yang, B. Lin et al, "Research of Quantum Genetic Algorithm and its application in Blind Source Separation", Journal of electronics, Jan. 2003, 62-68.

[8] P.S. Xia, "Quantum Computing", Computer research and development, 2001, Vol. 38: 1153-1168.

[9] G.X. Zhang, N. Li, et al, "A novel Quantum Genetic Algorithm and its application", Journal of electronics, 2004, Vol.32: 476-479.

Multi-objective Portfolio Optimization Utilizing Hybrid Genetic Algorithms

Jiabao Jiang, Wenbo Xu
Institute of Information Technology, Southern Yangtze University, Wuxi Jiangsu 214122, China
Email: jiabaojiang2004@sina.com or jiabaojiang2004@yahoo.com

ABSTRACT

Multi-objective portfolio optimization is decides the percentage of the overall portfolio value allocated to each portfolio component with specified risk and return and exchanging expense characteristics to make total investment risk and exchanging expense least, at the same time, make total investment return most and so on. The problem of multi-objective portfolio optimization is a problem of NP_hard. Ordinary methods are hard to be at the holistic best point. In this paper we study how to use Genetic Algorithms and Simulated Annealing to solve the problem of multi-objective portfolio optimization. We compare performance of single GA with performance of GA combined Genetic Algorithms. Many experiments that optimize the allocation of various stocks in the market of USA using Hybrid Genetic Algorithms and the analysis of experiment result indicate that the Hybrid Genetic Algorithms is a kind of efficient and reliable optimization arithmetic and it has determinate applied value in the field of Multi-objective portfolio optimization.

Keywords: Multi-objective, Portfolio, Genetic Algorithms, Simulated Annealing Algorithms

1. INTRODUCTION

Comparing with the investment of single asset, the merit of portfolio is that if we choose right assets and appropriate investing weight, total profit of an investment does not reduce whereas the investment venture is dispersed to every invested asset, namely the investment venture is diversified. The big companies of monetary funds management are responsible for the investment of trillions of dollars annually. This money is invested on different products like pension funds, banking insurance policies, stock exchange assets, and other series of financial assets. The selection of an appropriate investment portfolio is a basic process for these financial companies. Many of these decisions follow qualitative criteria, but nowadays decisions based on quantitative approaches are appearing.

When investing money, investors are interested in obtaining the maximum profit of an investment set, which also minimizes every kind of risks and expenses; moreover, the problem of portfolio has several constraints. Usually, the number of assets a portfolio can contain is fixed at a constant. In the same way, when an asset is selected to invest in, there is a minimum and maximum amount of possible investments for that asset. All of above-mentioned things show that the problem of investment portfolio is a problem of assembled optimization with several objectives and several constraints, at the same time, a problem of NP_hard. It is obviously a multi-objective project with several objective functions, some is maximum problems, some is maximum minimum problems. This article studies how to utilize genetic algorithms and simulated annealing algorithms to search the optimal solution of multi-objective investment portfolio.

Section 2 presents the complete mathematical model of the portfolio problem. Section 3 covers Genetic Algorithm and Simulated Annealing Algorithm for a multi-objective problem with three objective functions. Section 4 includes the implemented method for Genetic and Simulated Annealing Algorithm, The result obtained is shortly described and discussed in Section 5. Conclusions are exposed in Section 6. Finally, referenced literatures and bibliographies are given.

2. PORTFOLIO OPTIMIZATION

2.1 Multi-objective problems

Multi-objective problems are very common within the world of engineering optimization. This article optimizes a set of functions belonging to the different problem's characteristics [1,4,5]. Let us introduce a briefly mathematical description of a multi-objective optimization problem:
Let:

$x_1, x_2, \ldots x_n$ the variables of the problem
$f_1, f_2, \ldots, f_m$ the functions to optimize

We look for

$$\min \quad f_1(x_1,x_2,\ldots,x_n),\ldots,f_m(x_1,x_2,\ldots,x_n); \quad (1)$$

Subject to

$$\begin{aligned} &g_1(x_1,x_2,\ldots,x_n) \le b_1;\\ &g_2(x_1,x_2,\ldots,x_n) \le b_2;\\ &\quad\ldots\\ &g_r(x_1,x_2,\ldots,x_n) \le b_r; \end{aligned} \quad (2)$$

In such contexts, a set of functions belonging to the different problem's characteristics, hence, an optimal solution that make all functions $(f_1,f_2,\ldots,f_m)$ values least does not exist. At present, there is not an optimal method to solve a problem of assembled optimization with several objectives and several constraints. Our method is using appraising function.

Its basic thought is to construct an appraising function so that transform multi-objective problems to single objective problems in virtue of intuitional background of geometry and application, and then search the optimal solution by mature method of single objective optimization. In the course of execution Simulated Annealing Algorithm, we think that solution X* overmatch solution X, if X* and X Subject to following inequality:

$$\forall i \quad f_i(X^*) \le f_i(X)$$
$$\exists j \in 1,...,m \mid f_j(X^*) < f_j(X) \tag{3}$$

In the course of execution of GA, we construct appraising function by the method of optimal point. The method is that searching the current optimal value of every objective function: $f_1^*, f_2^*,..., f_m^*$ and then calculating the value of portfolio appraising function based on following equality:

$$MinF(X) = [\sum_{j=1}^{m}(f_j(X) - f_j^*)^2]^{1/2} \tag{4}$$

2.2 Mathematical modeling

There are several kinds of income and venture in the investment. To simplify the question, we compute related variables only according to the historical data

Let

M the total capital of the investors

N the available number of asset

K the number of assets to invest ($K \le N$)

b_i the purchasing quantum of the ith asset lowers than the fixed value

r_i the expected mean of the ith asset

p_i the price rate of exchange of the ith asset

n_i the minimum inversion ratio allowed in the ith asset

m_i the maximum inversion ratio allowed in the ith asset

w_i the actual inversion ratio allowed in the ith asset

z_i show whether asset i is invested ($z_i \in \{0,1\}$)

$c_i(w_i)$ the price of the exchange in the ith asset

$R_i(w_i)$ the net income of the ith asset

V_{ij} the covariance between the ith and the jth asset

The optimization problem can be formulated in the following way:

$$Min \quad Q = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j V_{ij} \tag{5}$$

$$Min \quad R = -\sum_{i=1}^{N} R_i(w_i) \tag{6}$$

$$Min \quad G = -\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \frac{R_i(w_i)R_i(w_i)}{V_{ij}} \tag{7}$$

$$R_i(w_i) = r_i \times M \times w_i - c_i(w_i) \times (1 + r_i) \tag{8}$$

$$c_i(w_i) = \begin{cases} 0, w_i = 0 \\ p_i \times b_i, 0 < w_i \le (1+p_i)\dfrac{b_i}{M} \\ \dfrac{p_i \times M \times w_i}{1+p_i}, (1+p_i)\dfrac{b_i}{M} < w_i \end{cases} \tag{9}$$

Subject to:

$$\sum_{i=1}^{N} w_i = 1, and, \sum_{i=1}^{N} z_i = K, and,$$
$$n_i * z_i \le w_i \le m_i * z_i, i = 1,..., N \tag{10}$$

2.THE INTRODUCTION OR THE ALGORITHMS

3.1 simulated annealing

Simulated annealing dates back to the statistical physics of the mid-1980s, between 1983 and 1985.Its physical foundation is the annealing of a solid, that is, the process of exposing a solid to high temperatures and leaving to cool down so slowly that, their particles look for the lowest energy positions [2].

Annealing can be considered an optimization problem as follows: a particle configuration is a solution to the problem, and a minimum energy configuration corresponds to an optimal solution. The energy of a configuration is the objective function value, and, last of all, the temperature is the search control parameter.

Using this equivalence, an algorithm able to accept solutions that worsen the objective function is designed. It makes use of a probability function based on the Boltzman distribution. Initially, when the temperature is high, this probability is also high. As the algorithm advances in its iterations, this probability tends to zero. On the limit, the algorithm behaves like a local search, only accepting solutions that unimproved the current one. In order to apply this technique, a series of annealing parameters must be defined, that is, an annealing scheme must be defined. Basically, an annealing scheme is composed of an initial and a final temperature, the temperature

updating function, and the number of iterations that are made with the same fixed temperature (the chain's size).

$chain$ The current iteration of the same fixed temperatures

$chain\_m$ the maximum of the iteration of the same fixed temperatures

$aleat$ the random number between 0 and 1

T_k current temperature value, for a given iteration k

H the updating parameter for the temperature T_k

Ω the searching space of the question

e one of the solutions of the question

$Q(e)$ the total venture of the solution computed by the formula (5)

$R(e)$ the total income of the solution e computed by the formula (6)

$V(e)$ the nearby solution sets of the solution e

The simulated annealing algorithm can be described as the pseudocode as follows:

$e_0 \in \Omega$

for (loops=0; loops < D; loops ++) {

for ($chain$ =0; $chain < chain\_m$; $chain$ ++){

Choose a solution $e \in V(e_0)$

$x = Q(e_0) - Q(e)$, $y = R(e_0) - R(e)$

if ((x>0 && y>=0) or (x>=0 && y>0))

Replace e_0 by e

else {

Choose $aleat \in [0,1]$

$a = x/T_k, b = y/T_k, m = e^{-a}, n = e^{-b}$

if $(y < 0 \,\&\, aleat < m) or\ (x < 0 \,\&\, aleat < n)$

Replace e_0 by e

}

if $V(e_0)$ has been selected, then reproduce $V(e_0)$

}

$T_{k+1} = H * T_k$

} Return (e_0)

The nearby solution sets $V(e)$ of the solution e is produced by the following pseudocode:

$e_0 \in \Omega$

inc = 0

Searching assets a and b with maximal vesture from the solution e ;

if $(R_a(w_a) < R_b(w_b))$ worst= a

else worst = b

v = e , c = w_{worst} / N

For (neighbors=0; neighbors<N; neighbors++) {

inc = inc + c

if ((w_{worst} - inc) $\geq n_{worst}$){

Searching assets k that is not worst from v

if((inc+ w_k)< m_k){ $w_k = w_k$ +inc, $w_k = w_k$ – inc}

else { $w_k = m_k, w_{worst} = w_{worst} - m_k + w_k$ }

Modify the optimal point: Q*、R*、G*,

Add v to $V(e)$;

}

else{

Searching assets k that is not v from all of assets,

$w_k = w_{worst}, w_{worst} = 0$

While ($w_k < n_k$)

{Searching assets m that is not k from v}

$x = (w_m - n_m) * aleat$

if $((x + w_k) < m_k)$ { $w_k = w_k + x, w_m = w_m - x$ }

else { $w_m = w_m - m_k + w_k, w_k = m_k$ }

Modify the optimal point's Q*、R*、G*, add v to V(e)

}

}

3.2 genetic algorithms

The evolution of the biology is a wonderful optimizing process. It generates fine species by selective elimination, mutation, gene heredity and so on .GA is a overall algorithm that come from the thought of biologic evolution .Its rationale is that a solution is expressed by a chromosome, and a chromosome is denoted by a string of binary code .In the begin of algorithm, a group of chromosomes are produced stochastically, then the adaptability of every chromosome is calculated [6]. According to the principle of the survival of the fittest, those fittest chromosomes are chose preferentially to give birth to offspring with good adaptability by intercrossing and aberrance. Such and such, the optimal solution or approximate optimal solution will be obtained .The particular manipulation is depicted in Section 4.

4. THE DESIGN OF MULTIOBJECTIVE PORTFOLIO OPTIMIZATION

In the process of the multi-objective portfolio optimization [1] [3] [4], our object of optimization is such as formula (4), (5), (6), so the evaluating function of a portfolio can be defined as:

$$Min: F(e) = [(Q(e) - Q^*)^2 + (R(e) - R^*)^2 + (G(e) - G^*)^2]^{\frac{1}{2}} \quad (11)$$

The approach of the realization of the hybrid Algorithms of GA and simulated annealing is

following:
Input:

The numbers of the assets selected, the popsize of the population, the initial value of the total investment, the history datum of the finance.

Output:

Optimal evaluating function value, the asset mark and the quantum of the investment, the price of the total exchange, the expected wealth of every asset, the venture of the total investment.

The first step: initialize the population randomly, and the initializing approach is as follows:

for (i=0; i< Popsize; i++) {

Select assets a_i of K to compose a kind of the investing assets @ randomly.

To each asset a_i of the subset @ of the asset aggregate, initializing investing weight w_{a_i} that satisfies $n_{a_i} \le w_{a_i} \le m_{a_i}$ randomly

Computing: S = $1-\sum_{i=1}^{K} w_{a_i}$

if ($S \neq 0$) {

for each $a_i \in @$

if (S>0) $f_{a_i} = m_{a_i} - w_{a_i}$

else $f_{a_i} = w_{a_i} - n_{a_i}$

}

for each $a_i \in @$

$$w_{a_i} = w_{a_i} + S * f_{a_i} \div \sum_{i=1}^{K} f_{a_i}$$

}

So composing a individual of the parent. Then using formulas (5),(6)and(7) to get Q_i ,R_i , G_i ,and get their optimizing points: Q*、R* and G* ;

The second step: evaluating every chromosome according to formula (11)

The third step: apply the fitness for the next generation, the approach is as follows:

For (i=0; i<Popsize; i++) {Select populations a and b from the parent according to (11) randomly.

Computing the evaluating function value F (a), F (b) of a and b.

If (F (a) > F (b))

Old [i] = a; that is: Apply parent a to crossover

Else

Old [i] = b; that is: apply parent b to mutation

}

The fourth step: generating the new generation by mutation and crossover as follows:

For (i=0;i<Popsize-1;i=i+2) //crossover

Copy parent [old [i]], parent [old [i+1]] to new [i],

New [i+1], and at a certain probability of crossover operate new[i], new[i+1] as follows:

Select X (1<X<K/2) weights of gene from new [i] and new [i+1] separately to exchange;

Select a asset mark of a gene from new [i] and new [i+1] to exchange;

For (i=0;i<Popsize;i++)//mutation

To mutate each gene j of the new [i] by a certain probability of mutation as follows:

Generate normal random number Y

$L=2\times m_j$

new[i].weight[j]= new[i].weight[j]+0.2×L×Y ×$[1/(1+times)]^{1/2}$//times are the current generation of evolution

The fifth step: According to (10), modify new [i] as

following pseudocode

if there is repeatable asset mark in new[i], select the new assets to replace the repeater randomly;

Do following operation to the every w_{a_j} of new [i]

If ($w_{a_j} < n_{a_j}$) $w_{a_j} = n_{a_j}$

If ($w_{a_j} > m_{a_j}$) $w_{a_j} = m_{a_j}$

Let $\sum_{j=1}^{K} new[i].w_{a_j} = 1$

least=0; most=0;

for (j=0; j<K; j++)

if ($w_{a_j} < n_{a_j}$)

if ($aleat < (n_{a_j} - w_{a_j}) / n_{a_j}$) {

Select new asset b_j

$w_{b_j} = w_{a_j}$, $w_{a_j} = 0$

if ($w_{b_j} < n_{b_j}$) { $w_{b_j} = n_{b_j}$

$least = least + n_{b_j} - w_{b_j}$ }

if ($w_{b_j} > m_{b_j}$){ $w_{b_j} = m_{b_j}$

$most = most + w_{b_j} - m_{b_j}$ }

else

$w_{a_j} = n_{a_j}$, $least = least + n_{a_j} - w_{a_j}$

$least = least - most$

While ($least > 0$) {Select a asset c_j of $w_{c_j} > n_{c_j}$ randomly from new[i]

x = $aleat * (m_{c_j} - w_{c_j})$

if ($(least + x) \le 0$)

$least = least + x, w_{c_j} = w_{c_j} + x$ }

Modify the optimizing points Q*,R* and G*

The sixth step: generate a nearby solution sets for each new [i];

The seventh step: simulated anneal for new[i]. according to formula (11) to compute the evaluating function value F(parent[i]) and F(new [i]) of parent[i] and new[i] at the probability

$$p = \frac{1}{1 + \exp(\frac{F(new[i]) - F(parent[i])}{T})}$$

Where T is the current temperature value. Replace child[i] by new[i] at the probability of (1 — P), Replace parent[i] by child[i].

The eighth step: if it meets the terminated condition; then return the optimal result. If not, return the third step.

5. EXPERIMENTAL RESULT

5.1 disposals of financial data

In order to validate the algorithm, we use USA data to test, which came from every week opening price, and closing price as well as the Market index between Dec.5, 1999 and Dec.5, 2005. First of all, according to following Property Fixed Price Model, we calculate each kind of stock anticipant repayment r_i (i=1, 2…N) based on these history data. Secondly, we calculate the covariance V_{ij} (i, j=1, 2,..., N) used to estimate the risk based on the stock anticipant repayments.

CAPM $r_i = 0.05+0.06\times d_i$;

In the Equality, 0.05 is set as the anticipant security rate of return, the coefficient 0.06 is set as the anticipant market rate of return, d_i originates from the property i history repayment average value.

5.2 enactments of algorithm parameters

Biggest iterative times of fixed temperature: $chain\_m$ = 100.

Initial temperature: T_0 =100.

Temperature T_k renewal parameter: H =0.95.

Simulation annealing iterative times: D=1000.

Heredity termination condition: the number of generations is 2000.

Popsize = 20.

The cross probability and the mutation probability may take pcross=0.4, pmutation=0.06 fixedly, in related procedure of this article is adjusted dynamically according to formula (12)

$$pcross = \begin{cases} \dfrac{F_{best}(e) - F(e_i)}{F_{best}(e) - F_{worst}(e)}; & F_{best}(e) \neq F_{worst}(e) \\ (rand()\%1000)/1000.0; \text{the rest} \end{cases}$$

$$pmutation = \min(pcross, 0.06); \quad (12)$$

F_{best} (e) and F_{worst} (e) denote separately the maximum and the minimum value of evaluating function in a current father generation. The expression $F(e_i)$ denotes the average value of two individual evaluating function values that is used to cross.

5.3 experimental results

The function is moved 50 times, and then we take its result to structure effective front. We obtain the following result.

Table 1. Procedure movement result with 5 assets

Algorithm	character	Minimum	Maximum
Genetic Algorithm	Risk	0.00024	0.0014
	Return ratio	0.00351	0.0104
Mixed Algorithm	Risk	0.000163	0.00018
	Return ratio	0.002626	0.002903

Table 2. Procedure movement result with 10 assets

algorithm	Character	Minimum	Maximum
Genetic Algorithms	Risk	0.000182	0.00140
	Return ratio	0.00305	0.01040
Mix Algorithm	Risk	0.000157	0.000183
	Return ratio	0.002592	0.00291

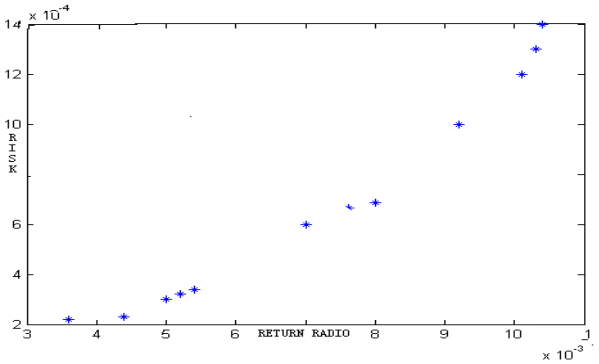


Fig. 1. The effective front of GA choosing 5 assets

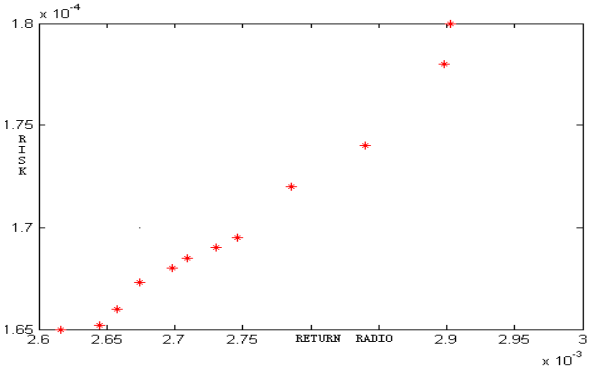


Fig. 2. The effective front of MA choosing 5 assets

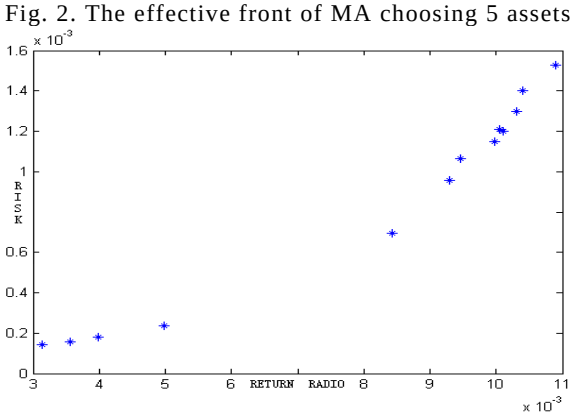


Fig. 3.The effective front of GA choosing 10 assets

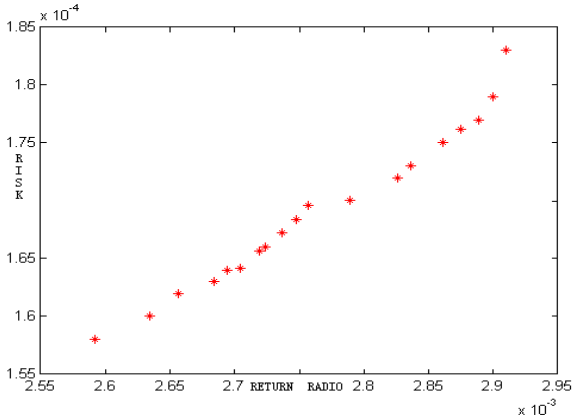


Fig. 4. The effective front of MA choosing 10 assets

It isn't difficult to see that result of the Mixed Algorithm of genetic algorithm and simulated annealing is more superior to the result of pure GA through the above tables and the graphs.

6. CONCLUSION

This paper elaborates genetic algorithm and the simulated annealing algorithm, and how to unify the simulated annealing algorithm and the genetic algorithm to constitute the multiobjective investment decision-making. By analyzing and comparing the results of pure genetic algorithm and mixed algorithm of simulated annealing algorithm and genetic algorithm we can safely draw the following conclusion:

Genetic algorithm's overall searching ability is superior to simulated annealing algorithm, but its local searching ability is inferior to simulation annealing algorithm. Mixed algorithm of simulation annealing algorithm and genetic algorithm has stronger searching ability and can be primly close to overall optimum point. Mixed algorithm of simulated annealing algorithm and genetic algorithm has very high practical application value in the multi-objective investment portfolio optimization domain.

7. REFERENCES

[1] A Multi-objective Approach to the Portfolio Optimization Problem; Ruben Armafianzas, Jose A. Lozano;

[2]S.Kirkpatrick, C.D.GelattJr etal,"Optimization by Simulated annealing," Science, vol.220:671-680, 1983.

[3] M.C CHAN1, C.C. WONG1, et al, Gordon Y-N Tang3.Genetic Algorithms in Multi-Stage Portfolio Optimization System 1Department of Computing, The Hong Kong Polytechnic University, Hong Kong.2GERAD and Ecole Polytechique de Montreal, Canada. 3Dept of Finance and Decision Sciences, Hong Kong Baptist University, Hong Kong.

[4] C.A. Coello, D.A.Van Veldhuizen, et al, Evolutionary Algorithms for Solving Multi-Objective Problems.NewYork: Kluwer Academic Publishers May 2002.

[5] T.J. Chang, N. Meade, et al, "Heuristics for cardinality constrained portfolio Optimization" Computers and Operations Research, 13(27): 1271-1302, 2000.

[6] K.C. Chalermkraivuth, S.B Gada, et al, "GE Asset Management, Genworth Financial, and GE Insurance Implement a Novel Sequential Linear Programming Algorithm for Portfolio Optimization," INFORMS Interfaces, (to appear in September 2005).

A New Dynamical Evolutionary Algorithm By Using Chaos

Xing Xu, Yuanxiang Li, Weiqin Ying
The State Key Laboratory of Software Engineering, Wuhan University
Wuhan, Hubei, 430072, P. R. China
Email: xx840614@163.com

ABSTRACT

In this paper, a new dynamical evolutionary algorithm using chaos (CDEA) is proposed based on the statistical mechanics. Using spread-spectrum characteristic of chaos, we generate multi-population in remote difference of the initial conditions. According to the selecting strategy of traditional dynamical evolutionary algorithm the best individual of the multi-population is migrated into the other populations replacing the worst one of them affirmatively has the chance to evolve. The algorithm presented in this paper by introducing chaos has bigger selective pressure, and can keep diversity of the system. In order to verify the effectiveness of our algorithm, we apply CDEA to solve the typical numerical function minimization problems. The experimental results show that CDEA is fast and reliable.

Keywords: Dynamical Evolutionary Algorithm, Statistical Mechanics, Chaos, Global Optimization, Multi-population.

1. INTRODUCTION

Dynamical evolutionary algorithm (DEA) is a new evolutionary computation technique developed in 2001 [1,2]. The DEA introduces a new selection strategy that the smaller variations and the fewer selected times of a particle; the more possible it is selected. The selecting mechanism of DEA drives all particles in the system moving, so it has smaller selective pressure. To speed up the convergence velocity, synchronously to avoid premature, we introduce chaos into DEA. Recently, chaotic sequences have been adopted instead of random ones and very interesting results have been shown in many applications such as secure transmission [3], natural phenomena modeling [4], neural networks [5], and nonlinear circuits [6]. Also in [7], chaotic time series were used in DNA computing procedures. From a time-series perspective, a chaotic system is characterized by signals with a broad-band spectrum that depend strongly on the initial conditions. Using spread-spectrum characteristic, we generate multi-population in remote difference of the initial conditions. So CDEA can keep the balance between diversity of populations and selective pressure.

The paper is organized as follows. In section 2, we give a short introduction on chaotic dynamics. In section 3, we offer a straightforward description of DEA .In section 4, we introduce the proposed CDEA. Then section 5, we make experiments to test CDEA. Finally, the conclusions are given in section 6.

2. CHAOS

Chaos is one of the most popular phenomena that exist in nonlinear system, and theory of chaos is one of the most important achievements of nonlinear system research. It is now widely recognized that chaos is a fundamental mode of motion underlying almost natural phenomena [8].Logistic equation Eq. (1) is brought forward for description of the evolution of biologic populations. It is the most common and simple chaotic function:

$$X_{n+1} = L * X_n (1 - X_n) \tag{1}$$

L is a control parameter which is between 0 and 4.0. For values of L in $(0,3)$, Eq. (1) will converge to some value x. For L between 3 and about 3.56 the solutions to Eq. (1) bifurcates into two, then four, then eight (and so on) periodic solutions. For L between 3.56 and 4 the solutions to Eq. (1) become fully chaotic: neither convergent nor periodic, but variable with no discernable pattern. When $L = 4$, the system is proved to be in chaotic state, given arbitrary initial value that is in $(0,1)$ but not equals with 0.25, 0.5 and 0.75, chaos trajectory will finally search non-repeatedly any point in $(0,1)$.

3. DESCRIPTION OF DEA

In general, we consider the Global Optimization Problems such as finding $x^* \in S$ such that

$$f(x^*) \le f(x), x \in S \tag{2}$$

where $S \subset R^d$ is compact, and $f : S \to R$ is a continuous function defined on S. Under these assumptions, the GOP is solvable, because f attains its minimum on S.

Suppose that a coding representation for solutions and evolutionary operators based on the code (mainly, crossover operator and mutation operator) are given. Now we take *N* coded solutions called particles, such as $x_1, x_2, ..., x_n$ to consist of a dynamical system. Correspondingly, their function values are $f(x_1), f(x_2), ..., f(x_n)$. An iterative step t in the DEA similarly like the generation in the traditional EA is called time *t*.

The momentum $p(t, x_i)$ of a particle x_i at time t is defined as where $f(t, x_i)$ denotes the function value of the particle x_i at time *t*. The activity $a(t, x_i)$ of a particle x_i at time t is defined as if x_i is selected to take part in an evolutionary operation at time *t*, otherwise keeps unchanging. Incorporating the momentum and the activity, a particle is assigned a selecting value defined as

$$slct(t, x_i) = \sum_{k=0}^{t} \left| p(k, x_i) \right| + a(t, x_i) \tag{3}$$

Generally, we can take a weight coefficient to indicate that which one of the two terms is more significant than another in selection, namely

$$slct(t, x_i) = \lambda \sum_{k=0}^{t} \left| p(k, x_i) \right| + (1 - \lambda) a(t, x_i) \tag{4}$$

In the DEA, $slct(t, x_i)(i = 1,2,..., N)$ are sorted in the order from small to large expressed as $slct(t)$ for short.

4. PROPOSED APPORACH

As outlined in the introduction, some applications of chaotic systems in EA and in optimization problems have been presented. In particular, in [9], the use of the chaotic Logistic function for the definition of a special mutation operator has been discussed, and some results have been presented showing how this chaotic mutation operator aids exploration in the search space. In this paper, sequences generated from chaotic systems substitute random numbers in all phases of DEA where it is necessary to make a random-based choice.

(1) During the creation of the initial population, the chaotic sequences which have remote difference on initial value, are used to generate the multi-population to preserve the diversity. Every several generations, the best individual of the multi-population is migrated into the other populations replacing the worst one of them. According to the Eq. (3) or Eq. (4), the migrated individual's $slct(t)$ is zero. So the best individual has the chance to evolve.

(2) In our program, we implement the crossover operation by this means. If we have chosen two parent particles X_1 and X_2, we take a number α generated by the chaotic sequences and then mapped into the interval $[-0.5,1.5]$ instead of $[0,1]$ for the purpose of expanding the searching area, including the boundary [10]. Then we can get two new particles X_1^* and X_2^* by linear combination of the two parents. That is,

$$X_1^* = \alpha X_1 + (1-\alpha) X_2 \quad (5)$$

$$X_2^* = \alpha X_2 + (1-\alpha) X_1 \quad (6)$$

(3) During the mutation algorithm, the chaotic sequences are used for replacing the individual which is chosen for mutation.

Therefore, chaotic sequences influence the behavior of all operators (mutation, crossover), not because new operators are introduced (as in [9]), but because all the existing standard operators work following the outcomes of a chaotic sequence instead of a standard random generator.

After preparing above, we illustrate the procedure of the CDEA below.

Procedure of CDEA

Initialize a particle dynamical system
t ← 0
N: the population size,
M: the number of populations,
MAXGENS: the number of max generations,
(a) generate the particle system $\Gamma_i = \{x_1, x_2, ..., x_N\}$
$i = 1,2,..., M$
(b) calculate function values of particles in Γ_i *and set* $p(t, x_i) = 0$, $a(t, x_i) = 0$ *for all* x_i *in* Γ_i *;*
(c) save the best particle and its function value in all the system Γ_i *;*
(d) calculate and sort $slct(t)$ *.*
Particle system evolution--Execute iteration for every Γ_i
repeat
(a) t ← t+1;
(b) select m particles on the forefront of $slct(t-1)$ *;*
(c) implement crossover operators introduced frontward on the m1 particles, mutation on m2 particles;
(d) modify function values, momenta and activities of the m(m=m1+m2) particles;
(e) save the best particle and its function value in the system Γ_i *;*
(f) every t0 generations, the best particle replace the worst one in the system Γ_i *;*
(g) modify and resort $slct(t)$ *;*
until{t>MAXGENS}.

5. NUMERICAL EXPERIMENTS

The significance of a new optimization algorithm depends on the effectiveness of solving practical problems. In this section, we use the CDEA to solve two typical hard global minimization problems, which are often used to test the performance and reliability of optimization algorithms. We will also compare the results with other methods.

Test 1: The first example is a simple quadratic function written as

$$f_1(X) = 100(x_1^2 - x_2)^2 + (1-x_1)^2$$
$$-2.048 \le x_i \le 2.048, i = 1,2 \quad (7)$$

Apparently, its global minimum value is 0 at the unique point $(1,1)$. This function is called as the De Jong's function number 1 for testing evolutionary algorithms.

Test 2: The second example is a function with some waves called the Bohachevsky function written as

$$f_2(X) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$
$$-50 \le x_i \le 50, i = 1,2 \quad (8)$$

This function has a global minimum value at point $(0,0)$.

Test 3: Six-hump camel back function (n=2)[11]

$$f_3(X) = (4 - 2.1_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$
$$-3 \le x_1 \le 3, -2 \le x_2 \le 2 \quad (9)$$

From [11], we know that the function has six local minima, just as the name of this function, and has one global minimum value -1.031628 at two points (-0.089842,0.712656) and (0.089842,0.712656). Because of this, many algorithms only can find the local minima, but loose the global optimization solutions.

Test 4: Shubert function III (n=2)[12]

$$f_4(X) = \{\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\}\{\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\}$$
$$+ (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2$$
$$-10 \le x_i \le 10, i = 1,2 \quad (10)$$

This function has 760 local minima and only one global minimum -186.7309 at the point (-1.42513,-0.80032). Because of the large number of local minima and the steep slope around the global minimum, this function has been wildly recognized as an important testing function.

In our programs the size of population is *N*=80, the number of population is *M*=4. At each iterative step, we define that *m1*=4, *m2*=2 and *t0*=50. That means 2 pairs of particles will take part in crossover, 2 particles will be mutated and every 50 generations, the best particle replace the worst one in the multi-population.

Table 1. Global optimal solutions of test functions

F	Opt-Solutions	Opt-Value
f_1	(1,1)	0
f_2	(0,0)	0
f_3	(-0.089842,0.712656) (0.089842,0.712656)	-1.031628
f_4	(-1.42513,-0.80032)	-186.7309

Table 2. Experimental Results

F	BEST	%	LEAST	AVER
f_1	(1.0,1.0) 0.0	100	154	1523
f_2	(0.0,0.0) 0.0	100	1051	2655
f_3	(-0.0898,0.7126) (0.0898,-0.7126) -1.031628	95	1451	6939
f_4	(-1.425,-0.8003) -186.7309	90	6172	21201

The program was executed 20 times for each problem. Table 2. summarizes the experimental results we obtained by using CDEA for those four functions. On Table 2., "BEST" means the best objective found by CEDA, "%" shows the percentage of successfully finding the solutions in 20 times. "LESAT" and "AVER" indicates the least and average number of the required iterations when finding a global optimum. Comparing Table 1. and Table 2., we can see that CDEA is quite an efficient optimization algorithm for GOP, and the successful rate of finding the optimal solutions is very high.

6. CONCLUSIONS

In this paper, attempting to balance the diversity of the population and selective pressure while searching for the optimal solutions, we proposed a new DEA based on the theory of statistical mechanics and Chaos. One of the aims of this work is to emphasize how coupling emergent results in different areas, like those of evolutionary computation and complex dynamics, can improve the search for solutions in some optimization problems.

By solving some typical testing problems, the efficiency and good performance of the DEA are tested. Although we get some achievement though this work, we still have some problems to research. They are quantificationally analyzing selective pressure [13, 14] of DEA, and how to use the DEA in many other fields, such as combinatorial optimization problems, where there are many difficult problems existing. These are all the problems left for further studies.

After all, putting use of some theories in natural science into computation has an extensive prospect. For the long run, we should pay much more attention on this field, and may get more and more useful and interesting conclusions.

REFERENCES

[1] Y.X.Li, X.F.Zou et al, "A new dynamical evolutionary algorithm from statistical mechanics," Chinese Journal of Computer and Science Technology, 2002.

[2] X.F.Zou, Y.X.Li et al, "Finding global minima with a new dynamical evolutionary algorithm," Wuhan University Journal of Natural Sciences, 2002.

[3] R. Caponetto, M. Criscione et al, "Synthesis of a programmable chaos generator, based on CNN architectures, with applications in chaotic communication," in Proc .CNNA '98, London, U.K., Apr. 14–17, 1998, 124–129.

[4] M. Bucolo, R. Caponetto et al, "How the chua circuit allows to model population dynamics," presented at the Proc. NOLTA '98, La Regent, Crans-Montana, Switzerland, Sept. 14–17, 1998.

[5] H. Nozawa, "A neural network model as globally coupled map and application based on chaos," Chaos, vol. 2, 1992, 377–386.

[6] P. Arena, R. Caponetto et al, "Self organization in non recurrent complex system," Int. J. Bifurcation and Chaos, 2000, 10（5）： 1115–1125.

[7] G. Manganaro, J. Pineda de Gyvez, "DNA computing based on chaos," in Proc. 1997 IEEE International Conference on Evolutionary. Piscataway, NJ: IEEE Press, 1997, 255–260.

[8] B.Li, W.S.Jiang, "Chaos optimization method and its application", Control Theory and Applications, 1997, 14（4）：63-615.

[9] J. Determan, J. A. Foster, "Using chaos in genetic algorithm," in Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, vol. 3, 1999, 2094–2101.

[10] T.Guo, L.S.Kang, "A new Evolutionary Algorithm for Function Optimization", Wuhan University Journal of Nature Science, vol. 4, 1999, 409-414.

[11] Z. Michalewicz, "Genetic algorithms + Data structures= Evolution programs", Springer-Verlag, Berlin, Germany 1996.

[12] R. Ge, "The globally convex zed filled functions for global optimization", Applied Mathematics and Computation. Vol.35, 1990, 131-158.

[13] D.E. Goldberg, K. Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithm", In Gregory Rawlins, editor. Foundations of Genetic Algorithms, San Mateo, CA: Morgan Kaufmann Publishers, 1991, 69-93.

[14] T. Bäck, "Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms", in Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE Press, Piscataway, NJ, 1994, 57-62.

Solving Combinatorial Optimization Problem Using Quantum-Behaved Particle Swarm Optimization

Na Tian, Jun Sun, Wenbo Xu
School of Information Technology, Southern Yangtze University,
No. 1800, Lihudadao Road, Wuxi, Jiangsu 214122, China

ABSTRACT

There are many combinatorial optimization problems for which there exists no direct or efficient method of solution. In this paper, we proposed a novel algorithm called Quantum-Behaved Particle Swarm Optimization (QPSO) that bases rank for solving combinatorial optimization. The experimental results show that it can achieve good results.

Keywords: Quantum-Behaved Particle Swarm Optimization; Traveling Salesman Problem; Combinatorial Optimization

1. INTRODUCTION

In the fields of Operation Research and Artificial Intelligence, there are many combinatorial optimization problems for which there exists no direct or efficient method of solution. An unguided search is extremely inefficient since many of these problems are NP-complete. Thus a good exact algorithm for its optimal solution in polynomial time is unlikely to exist. Therefore, optimal solution strategies must be scarified in favor of fast heuristic techniques.

In this paper, we look at one representative combinatorial optimization problem—Traveling Salesman Problem. We choose the traveling salesman problem because it is representative problem of a wide variety of combinatorial optimization problems where the solution space is all permutations of n objects. Other combinatorial optimization problems that fall into this category include the Bin Packing Problem, Job Scheduling, Stock cutting, Vehicle Routing and Transportation Scheduling Problems, etc. Developing efficient algorithms to the TSP problem will have direct applications for solving a host of other practical combinatorial optimization problems. By now, TSP has been studied by many metaheuristic approaches such as nearest neighborhood search (NNS), simulated annealing (SA), tabu search (TS), neural networks (NN), Ant Colony System (ACS)[1], and genetic algorithm (GA)[2]. QPSO has been proven to succeed in continuous problems, much work has been done effectively in this area, but for discrete problems, it is a new field.

PSO is an algorithm that follows a collaborative population-based search model [2-7]. Each individual of the population, called a 'particle', flies around in a multidimensional search space looking for the optimal solution. Particles, then, may adjust their position according to their own and their neighboring-particles experience, moving toward their best position or their neighbor's best position. In order to achieve this, a particle keeps previously reached 'best' position in a cognitive memory. PSO performance is measured according to a predefined fitness function (cost function of a problem). Balancing between global and local exploration abilities of the flying particles could be achieved through user-defined parameters. PSO has many advantages over other heuristic techniques such that it can be implemented in a few lines of computer codes, it requires only primitive mathematical operators, and it has great capability of escaping local optima.

Inspired by the analysis of convergence of PSO, we study the individual particles of a PSO system moving in a quantum multi-dimensional space [9], and the experiment results indicate that the QPSO works better than standard PSO on several benchmark functions and it's a promising algorithm.

The remainder of the paper is organized as follows. Section 2 describes the QPSO algorithm. Section 3 describes our QPSO algorithm for the traveling salesman problem. In Section 4, we present the experimental results and the discussions. Finally, a summary is given in Section 5.

2. QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION

In the quantum model of a PSO, the state of a particle is depicted by wave-function $\Psi(\bar{x}, t)$, instead of $\bar{x}$ and $\bar{v}$. The dynamic behavior of the particle is widely divergent from that of the particle in traditional PSO systems in that the exact values of $\bar{x}$ and $\bar{v}$ cannot be determined simultaneously. We can only learn the probability of the particles appearing in position $\bar{x}$ from probability density function $|\Psi(X,t)|^2$, the form of which depends on the potential field the particle lies in.

In [9], Jun Sun et al. employed Delta potential well with the center on point $\bar{p} = (p_1, p_2, \ldots p_n)$ to constrain the quantum particles in PSO in order that the particle can converge to their local $\bar{p}$ without explosion. The wave-function of the particle in Delta potential well is as follows:

$$\Psi(x) = 1/\sqrt{L} \exp(-\|p-x\|/L) \quad (1)$$

And the probability density function is

$$Q(x) = |\Psi(x)|^2 = \frac{1}{L} \exp(-2\|p-x\|/L) \quad (2)$$

The parameter L depending on energy intension of the potential well specifies the search scope of a

particle and is called “Creativity” or “Imagination” of the particle in this paper.

In quantum-behaved PSO, search space and solution space of the problem are two spaces of different quality. Wave-function or probability function of position depicts the state of the particle in quantized search space, not informing us of any certain information about the position of a particle that is vital to evaluate the fitness of a particle. Therefore, state transformation between two spaces is absolutely necessary. In terms of quantum mechanics, the transformation from quantum state to classical state is called collapse, which in nature is the measurement of a particle’s position. Monte Carlo Method, a stochastic simulation, can realize the process of measurement on computers. And the position can be given by

$$X(t)=p \pm \frac{L}{2}\ln(1/u) \quad (3$$

In [11], the parameter L is evaluated by

$$L(t+1)=2*\alpha*\left|p-x(t)\right| \quad (4)$$

Thus the iterative equation of Quantum PSO is

$$X(t+1)=p \pm \alpha*\left|p-x(t)\right|*\ln(1/u) \quad (5)$$

Where, $p_d = (\varphi_1 * p_{id} + \varphi_2 * p_{gd})/(\varphi_1 + \varphi_2)$

α is the only parameter in the algorithm is Creative Coefficient.

u is a uniform random value in [0, 1]

The algorithm shows below:

QPSO Algorithm

Initialize population: random x_i

Do

For i=1 to population size M

If $f(x_i) < f(p_i)$ *then* $p_i = x_i$

$p_g = min(p_i)$

for d=1 to dimension D

$fi_1 = rand(0, 1), \quad fi_2 = rand(0, 1)$

$p=(fi_1 * p_{id} + fi_2 * p_{gd})/(fi_1 + fi_2)$

u=rand (0,1)

if rand(0, 1)>0.5

$x_{id} = p - \alpha * abs(p - x_{id}) * ln(1/u)$

else

$x_{id} = p + \alpha * abs(p - x_{id}) * ln(1/u)$

Until termination criterion is met

3. Quantum-Behaved Particle Swarm Optimization for Traveling Salesman Problem

In this Section, we describe the formulation of a QPSO algorithm for the traveling salesman problem. In this paper, we consider the cities are in the plane with Euclidean distance.

One of the key issues in designing a successful PSO algorithm is representation step, i.e. finding a suitable mapping between problem solution and PSO particle. In this paper, we setup a search space of N dimension for an N city problem. Because the city number is discrete value, we give every city a rank, which has continuous value. The cities are permutated according to their ranks. For example, one candidate solution {54.3, 32.5, 67.0, 22.2, 33.3} represents the permutation {3, 1, 5, 2, 4}. In this way, we can translate the discrete optimization problem to continuous optimization problem. The algorithm starts by generating randomly the population. We can define a up-bound and down-bound for the ‘position’ of the particle. We use use Eq. [5] as the ‘position’ update function. After the positions of all the particles have updated, we calculate the fitness according the cities’ new ranks. Finally, the algorithm repeats this whole process of determining the global and the local best positions, updating particle position and evaluating new particle position until a user-determined criterion is satisfied. In our case, we mapped this criterion to a maximum number of generations.

4. Experimental results

We use a TSP beanchmark problem with 14 nodes to test the validity of our approach.

Table 1. Description of TSP (14 nodes)

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**
CoordX	**16.47**	**16.47**	**20.09**	**22.39**	**25.23**	**22.00**	**20.47**
CoordY	**96.10**	**94.44**	**92.54**	**93.37**	**97.24**	**96.05**	**97.02**
Node	**8**	**9**	**10**	**11**	**12**	**13**	**14**
CoordX	**17.02**	**16.30**	**14.02**	**16.35**	**21.52**	**19.41**	**20.09**
CoordY	**96.29**	**97.38**	**98.12**	**97.38**	**95.59**	**97.13**	**94.55**

The TSP problem is described as Table 1. Fig. 1 shows the initial random solution, and Fig. 2 shows the best solution we have got.

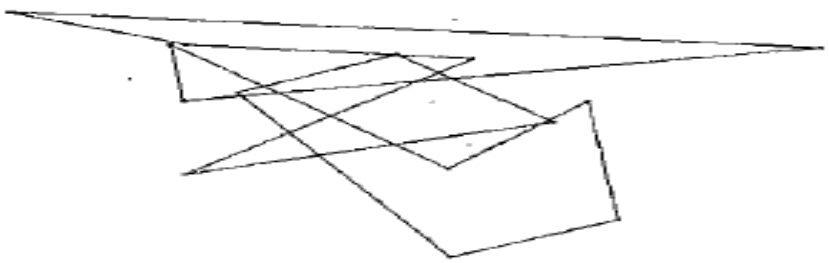


Figure 1 The initial random solution. Cost: 59.8462

Figure 2 The best solution we have got. Cost: 30.8785

Performance of the experiment:
The size of the solution space: 14!
The number of the iterations of the algorithm: 20000.
The size of search space: 2000000.
Search space/ solution space: 0.064%.
The best solution we have got:
1->10->9->11->8->13->7->12->6->5->4->3->14->2
Cost: 30.8785 (Equal to the optimal solution)[10].

5. Conclusion

In combinatorial optimization fields, traveling salesman problem is the representation of many real world problems. In this paper, a novel heuristic algorithm called QPSO algorithm is proposed for the traveling salesman problem. The result we got is the same as the optimal solution. Moreover, QPSO applications to other combinatorial optimization problems need further investigation.

6. REFERENCE

[1] L. Huang, CH.G. Zhou, et al, Hybrid Ant Colony algorithm for Traveling Salesman Problem. Progress In Natural Science. April 2003, 13 (4).
[2] C.G. Zhou, Computing Intelligence (in Chinese) Changchun: Publishing House of Jilin University, 2001.
[3] K. James, R. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks (December)(1995) 1942-1948.
[4] K. James, E. Russell, et al, Swarm Intelligence, Morgan Kaufmann, San Mateo, CA, 2001.
[5] Y. Shi, R. Eberhart, Empirical Study of Particle Swarm Optimization, Proceedings of the Congress on Evolutionary Computation (CEC99)(July), 199, pp.1045-1950.
[6] P.N. Sugantha, Particle Swarm Optimization with Neighborhood Operator, Proceedings of the Congress on Evolutionary Computation (CEC99)(July), 1999, pp.1958-1962.
[7] E. Ozcan, C.k. Mohan, Particle Swarm Optimization: Surfing the Waves, Proceedings of the Congress on Evolutionary Computation (CEC99) (July), 1999, pp.1939-1944
[8] M. Clerc, The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization, Proceedings of the Congress on Evolutionary Computation (CEC99) (July), 1999, pp.1951-1957.
[9] J. Sun, B. Feng, et al, "Particle Swarm Optimization with Particles Having Quantum Behavior" IEEE Proc. of Congress on Evolutionary Computation, 2004.
[10] K.P. wang, Particle Swarm Optimization for Traveling Salesman Problem. Preceeding s of the Second International Conference on Machine Learing and Cybernetics, Xi'an, 2-5 November 2003

Automatic Recognition of Fabric Weave Patterns Using Texture Orientation Features *

Jianqiang Shen1,2 ,Zhaofeng Geng
1College of Information Science and Technology ,Donghua University
Shanghai, 200051, China
Email: jianqiangshen@126.com
And
Xuan Zou, Yongbin Pan
2Shanghai -Hamburg Joint College , University of Shanghai for Science and Technology
Shanghai, 200031, China
Email: zouxuan@citiz.net

ABSTRACT

A novel approach is proposed for detecting fabric texture orientations and recognizing weave patterns. Wavelet transform is suited for fabric image decomposition and Radon transform is fit for line detection in fabric texture. Their excellent performances are combined to detect texture orientations in this study. Since different weave patterns have their own regular orientations in the original image and sub-band images composed by Wavelet transform, these orientations features are extracted, and then used as SOM inputs to achieve an automatic recognition of fabric weaves. The contribution of this algorithm is that it not only identifies basic fabric weaves but also accurately classifies some derivative twill weaves such as angular twills and pointed twills. The experimental results show that the proposed method is feasible and effective.

Keywords: Wavelet Transform, Radon Transform, Fabric weave pattern, SOM.

1. INTRODUCTION

The mechanical properties of a woven fabric depend not only on those of the yarns constituting the fabric, but also on the structural properties of the fabric itself, such as weave pattern, yarn number, fabric density, and yarn crimp. At present, the analysis of fabric weave structure and fabric pattern recognition depends largely on human inspection. The results in some cases, however, are not very reliable with the naked human eye. Therefore, it is highly desirable to develop automatic analysis techniques to identify fabric characteristics. Image processing technologies have been proved to be an efficient method of analyzing fabric structures and identifying fabric weave patterns [1,2,3]. Image Processing Technology to achieve an automatic recognition of fabric weave parameters has been studied since the middle of the 1980s. The main methods so far are Fourier transform techniques for identifying weave patterns. A number of sources in the literature note the use of frequency domain image processing [4]. The major principle here is the peak in the power spectrum image, representing frequency terms of periodic elements from which basic weave patterns can be recognized. But similar weave patterns have similar power spectra, making it difficult to distinguish them. Using information of warp and weft floats is another way to determine weave patterns [5, 6]. The main principle here is to first locate warp and weft crossed areas by analyzing gray value changes in both horizontal and vertical directions, and then use these areas' geometric shapes to determine warp floats or weft floats. However, due to differences in yarn material, count and density, different fabrics have diverse geometric shapes for warp and weft floats. This means the criteria used for one fabric may not be proper for another fabric. Another tool for frequency domain analysis is the two-dimensional discrete wavelet transform (2DDWT), which has the advantage of providing information about the data that are localized in both frequency and location [7]. In recent years, the algorithms based on various artificial neural networks have been widely used for fabric pattern recognition [8]. However, most of algorithms presented in the literature only focus on the automatic recognition of the three basic weave patterns.

In this study, an effective method based on Wavelet and Radon transform are developed to extract texture orientation features from fabrics, and then a self-organized map neural network is used for classifying woven fabric patterns. On one hand, Wavelet transform is suited for fabric image decomposition and different weave patterns have their own regular orientations in the original image and sub-band images composed by Wavelet transform. On the other hand, Radon transform is fit for line detection in fabric texture. Therefore, texture orientations can be used as features to effectively identify fabric weave patterns. We put their excellent performances together to extract texture orientation features from the original image and sub-band images. These effective texture orientation features are used as SOM inputs to achieve an automatic recognition of fabric weave. The main contribution of this algorithm is that it not only identifies three fundamental fabric weaves but also accurately classifies some derivative twill weaves such as angular twills and pointed twills. The experimental results show that the proposed method is feasible and effective.

2. WAVELET, RADON TRANSFORM AND SOM

2.1 Wavelet Transform

There are efficient algorithms for computing the 2DDWT that yield a fast wavelet transform analogous to the fast Fourier transform. Wavelet transform is a reliable and effective approach for Fabric image decomposition. The decomposition of wavelet transform is the excellent way to present warp and weft information from a fabric image, since the fabric image can be decomposed in different directions and Frequencies. Fig. 1. shows one stage of the

2DDWT for a pre-processed fabric image. Here, DWT is implemented with a bank of 1D low-pass and high-pass analysis filters. After one stage of decomposition, an image at resolution level 1 will be decomposed into four sub-band images and the three details images, correspond to the low-high, high-low and high-high bands in the frequency domain respectively. The remaining (low-low) component is a low-pass filtered version of and can be used for further sub-band decomposition. A 2DDWT with d decomposition levels will thus result in a total of 3d+1 sub-bands. On the other hand, the inverse DWT can be used to inversely reconstruct the image according to the sub-band images. The approximation and detail coefficients can be used to reconstruct the image based on the wavelet at the current analysis scale, and the details provide a measure of the "fit" of the wavelet with the image data in the particular orientation (horizontal, vertical, or diagonal) at the current analysis scale.

We used the Matlab software package to perform the 2DDWT analysis in this study. This package includes a Wavelet Toolbox that provides a range of pre-defined wavelets that can be used as the basis for analysis.

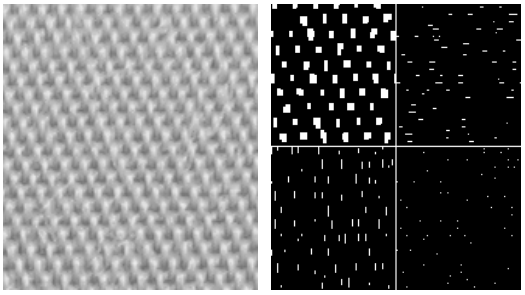


Fig. 1. One stage of the 2DDWT for a pre-processed fabric image

2.2 Radon Transform

Radon transform was performed with the Matlab Image Processing Toolbox. The radon function in the Image Processing Toolbox computes projections of an image matrix along specified directions. A projection of a two-dimensional function *f*(*x*,*y*) is a set of line integrals. The radon function computes the line integrals from multiple sources along parallel paths, or beams, in a certain direction. The beams are spaced 1 pixel unit apart. To represent an image, the radon function takes multiple, parallel-beam projections of the image from different angles by rotating the source around the center of the image. The following figure shows a single projection at a specified rotation angle.

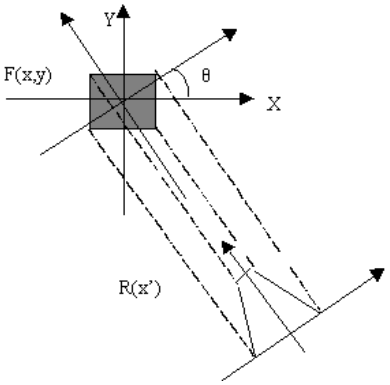


Fig. 2. Radon transform

Projections can be computed along any angle θ. In general, the Radon transform of *f*(*x*,*y*) is the line integral of f parallel to the y´ axis, which can be defined as follows:

$$R_0(x') = \int_{-\infty}^{\infty} f(x'\cos\theta - y'\sin\theta, x'\sin\theta + y'\cos\theta)dy' \quad (1)$$

where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

The radon function computes the Radon transform, which is the projection of the image intensity along a radial line oriented at a specific angle. *R* = radon (*I*, *theta*) returns the Radon transform of the intensity image *I* for the angle *theta* degrees. If *theta* is a scalar, the result *R* is a column vector containing the Radon transform for *theta* degrees. If *theta* is a vector, then *R* is a matrix in which each column is the Radon transform for one of the angles in *theta*. [*R*, *Xp*] = radon (...) returns a vector *Xp* containing the radial coordinates corresponding to each row of *R*. The radial coordinates returned in *Xp* are the values along the *x* ′-prime axis, which is oriented at *theta* degrees counter clockwise from the x-axis. The origin of both axes is the center pixel of the image, which is defined as floor ((size(*I*)+1)/2), this is the pixel on the *x*´-axis corresponding to *x* ′=0.

2.3 SOM-based clustering

The self-organizing map algorithm developed by Kohonen [9] has been successfully applied to various problem domains such as cluster analysis, speech recognition, etc. .The example architecture self-organizing map is shown in Fig. 3.In general, the self-organizing map can capture the topology and distribution of the input data and provide a clustering analysis. The architecture of SOM includes two layers: input layer and Kohonen layer. The Kohonen layer can be one or two dimension. The learning process is competitive and unsupervised, meaning that no teacher is needed to define the correct output (or actually the cell into which the input is mapped) for an input. The input layer receives the input feature vector, and weights between neurons and input feature vectors are calculated. The neuron with minimum Euclidean distance is called the winner neuron. Through competitive learning, the weights of the winner neuron and its neighborhood neurons are updated. In the basic version, only one map node (winner) at a time is activated corresponding to each input. The locations of the responses in the array tend to become ordered in the learning process as if some meaningful nonlinear coordinate system for the different input features were being created over the network.

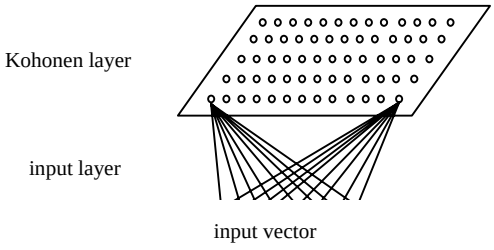


Fig. 3. Architecture of SOM

3. METHODOLOGY

We used the Matlab software package to perform the

2DDWT analysis, Radon Transform and clustering analysis with SOM algorithm. The proposed method can be described as the following steps:

Step 1: Fabric image capture and pre-processing

An appropriate alignment of yarns parallel and perpendicular to image axes is necessary for this algorithm to work properly, so the fabric has to be placed carefully before the images of prepared fabric samples are captured with CCD camera. Then the captured images are transferred into intensity images. The intensity of an image refers to a two-dimensional light intensity function, denoted by I(x, y). For the images, the intensity value at coordinates (x, y) is the gray level at that point lies in the range of (0,255), 0 for black and 255 for white. Finally, a series of image pre-processing steps are taken. These steps include Gaussion filtering, image contrast stretching (histogram equalization), thresholding and morphological operations (such as erosion and dilation to eliminate unwanted noise pixels) [10] and so on. Compared to raw fabric images, the characters of pre-processed images become more clearly visible.(see Fig. 4.)

Step 2: Fabric image decomposition with Wavelet transform

The simplest wavelet is the Haar wavelet, which has the general appearance of a square wave, and it is suggested as an analysis basis for data with "jump" or "step" features [11], as one would expect to find in the image data from the repeating pattern of a fabric. Analysis with the Haar wavelet is also computationally simpler than many other wavelets [12]. After one stage of processing, one fabric image $I(x,y)$ is decomposed into four frequency band sub-images described as above. Then the horizontal and vertical sub-band images are reconstructed into the images $I_V(x,y)$ and $I_H(x,y)$ respectively with Matlab wavelet tool. Both images have the same size as the original image.

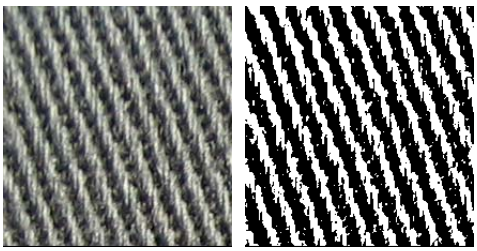


Fig. 4. Pre-processed image

Step 3: Texture orientation detection with Radon transform

We can find the fact that the images: I_V(x,y)and I_H(x,y) ,as well as original image $I(x,y)$ have their own unique texture orientations corresponding to different fabric weaves. Therefore, texture orientations in I_V(x,y), $I_H(x,y)$ and $I(x,y)$ can be extracted as features to effectively identify different fabric weaves. Radon Transform is applied to detect fabric texture angles in these images respectively. Fig. 5. illustrates the projection result of pre-processed image shown in Fig. 4. The final fabric texture orientation α is defined by the following set of conversion equations according to the different projection angle θ.

$$\alpha = \begin{cases} \theta + \dfrac{\pi}{2} & if \quad 0 \le \theta < \dfrac{\pi}{2} \\ \theta - \dfrac{\pi}{2} & if \quad \dfrac{\pi}{2} \le \theta < \pi \end{cases} \tag{3}$$

Step 4: Feature representation

Texture orientations measured in I_V(x,y), $I_H(x,y)$ and $I(x,y)$ by Step 3 are used as the input vector in the artificial neural network to recognize the weave pattern, since there is the close relation between these texture orientations and fabric weaves. In this study, we extracted five features (angles) at most from I_V(x,y), $I_H(x,y)$ and $I(x,y)$ respectively, and obtain fifteen orientation features in total. The input feature vector I can be represented as follows:

$$I = (\alpha_i^V, \alpha_i^H, \alpha_i) \quad \text{and} \quad i \in 1,2,3,4,5 \tag{4}$$

where α_i^V, α_i^H and α_i denote texture angles detected in I_V(x,y), $I_H(x,y)$ and $I(x,y)$ respectively.

For instance, there are four texture angles: 30º, 90º, 135º,175º in Fig. 7.(b)(plain weave sample). The fifth angle is set as 0, which indicates no fifth angle existing. Each angle feature in the vector can be easily normalized to [0,1] by Eq.(5) before they are fed into the SOM algorithm.

$$\alpha_i = \alpha_i / 180 \tag{5}$$

Therefore, α_i can be regarded as (30/180, 90/180, 135/180, 175/180, 0). The normalization results in the fact that each feature has the same importance in the SOM algorithm.

Step 5: SOM-based clustering

We use self-organizing map artificial neural network algorithm with a two-dimensional neural array of size 4×5 to automatically recognize the fabric pattern. The target is divided into several classes contain plain, satin and some derivative twill weaves such as angular and pointed twill. The orientation feature vector I formed in Step 4 is used as SOM inputs. The neural networks are trained by the SOM algorithm to classify the fabric pattern. After training and validation, the SOM net classifies an input vector by assigning it to the same class as the output unit that has its weight vector closest to the input vector.

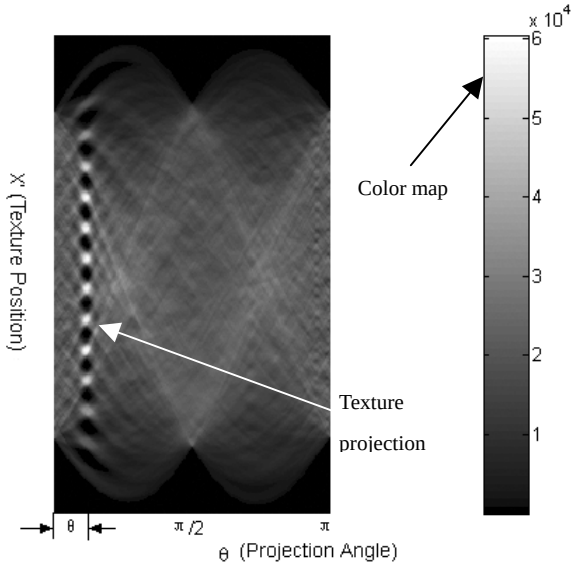
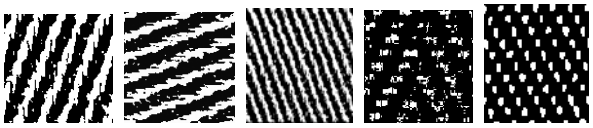


Fig. 5. The projection result of pre-processed image

4. EXPERIMENTAL RESULTS

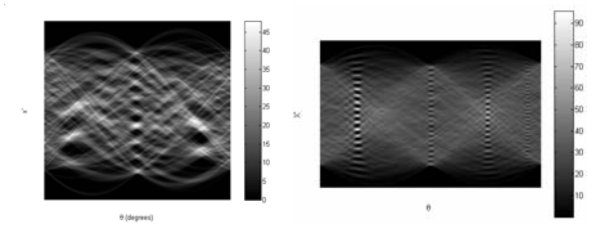
The horizontal yarn (weft) and vertical yarn (warp) weave each other, which forms a fabric. Fabric can be divided into three fundamental weaves, which are plain, twill and satin weave. Twill contains some derivative weaves such as angular twills and pointed twills. The experiments have been

conducted on various fabric samples to detect fabric texture angles with the proposed method. To assess the effectiveness of proposed method, the experimental samples cover all the kinds of fundamental weaves. The fabric texture angles of samples (see Fig. 6.) are measured, and part of experimental results is listed in Table 1. The experimental results show that the fabric texture angles of samples detected coincide well with those of manual operations. Meanwhile, we can find the fact that the projections computed with Radon transform on the original image and each sub-image from 2DDWT is unique for different fabric weaves. For example, original image of pointed twill has the similar Radon transform result to the original image of plain (see Fig. 7.), but orientation features in Radon transform resulting from their horizontal images have apparent differences (see Fig. 8.). In other words, effective features in various weave patterns can be extracted from regular orientations existing in a fixed kind of fabric weave. We have tested the vector form of these features with the SOM algorithm to recognize the unknown pattern. In this experiment, the experimental materials also include plain, twill, and satin woven fabrics. The input feature vector is mapped to Kohonen output layer and position of output neuron represents only one the fabric pattern. The SOM neural network properly adapt in detail within 150 epochs with the default parameters provided by Matlab neural network toolbox. Our classification results of an unknown test fabric show good agreement with manual measurements.



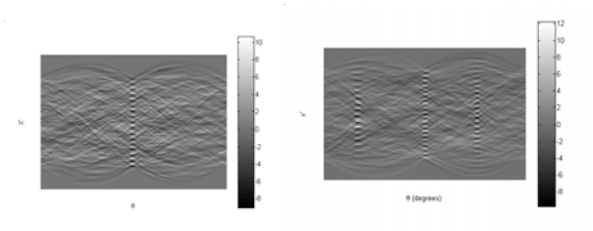
(a) twill 1 (b) twill 2 (c) twill 3 (d) pointed twill (e) plain

Fig. 6. Experimental materials



(a) pointed twill (b) plain

Fig. 7. Radon transform results of original images



(a) pointed twill (b) plain

Fig. 8. Radon transform results of horizontal images

Table 1. Part of experimental results (unit: degree)

sample	θ_1	θ_2	α_1	α_2	weave
Fig. 6.(a)	155	180	65	180	left twill
Fig. 6.(b)	115	180	25	180	left twill
Fig. 6.(c)	25	180	115	180	right twill
Fig. 6.(d)	30	150	120	60	pointed twill
Fig. 6.(e)	0	85	90	175	plain

5. CONCLUSIONS

This paper has applied Digital Image Processing and other advanced technologies such as 2DDWT and neural network to achieve automatic recognition of fabric weave patterns. The advantage of this algorithm is that it not only classifies three fundamental weave types but also accurately recognizes some derivative weaves. Various experiments have been conducted to demonstrate the proposed method is feasible and effective. First, the 2DDWT yields an approximation of the original image and three sets of details that represent the horizontal, vertical, and diagonal details in the original image. Then we can reconstruct each sub-image in the particular orientations (horizontal and vertical) at the current analysis scale. After that, projections of an image matrix along the specified orientations (angles) are computed with Radon Transform on the original image and each sub-image. Since the projection information contains close relations with different fabric weave patterns, orientation features are effectively extracted from the fabric original image and each sub-image. Finally these features are fed into the SOM neural network algorithm to classify woven fabric patterns. The experimental results show that the proposed method is feasible and effective.

REFERENCES

[1] C.Huang, S.Liu, et al, “Woven Fabric Analysis by Image Processing, Part I: Identification of Weave Patterns,” Textile Research Journal, 2000, 70(6)： 481-485.

[2] Kang, T. J., Kim, et al, “Automatic Recognition of Fabric Weave Patterns by Digital Image Analysis”, Textile Research Journal, 1999, 69 (2): 77-83.

[3] Nishimatsu, T., “Automatic Recognition of the Interlacing Pattern of Wool Fabrics”, Part 1: Determination of the Pattern by Using Image Information Analysis, Journal of Textile Mach. Soc. Jpn., 1991,44(6) ： 51-57.

[4] B.Xu, “Identifying Fabric Structures with Fast Fourier Transform Techniques”, Textile Research Journal, 1996, 66(8)：496-506.

[5] Tae Jin Kang. “Automatic Recognition of Fabric Weave Patterns by Digital Image Analysis”, Textile Research Journal, 1999, 69 (2): 77-83.

[6] Tae Jin Kang. “Automatic Recognition of Fabric Weave Patterns by Digital Image Analysis”, Textile Research Journal, 2001,71(3): 261-270.

[7] Hubbard, B. B., The World According to Wavelets, Wellesley, MA: A. K. Peters, Ltd., 1996.

[8] F.K.Wang, T.Chang, et al, “Fabric Classification Based on Recognition Using a Neural Network and Dimensionality Reduction” , Textile Research Journal , 1998, 68(3): 179-185.

[9] T. Kohonen. Self-Organization and Associative

Memory, 3rd ed., New York, Berlin: Springer-Verlag, 1989.

[10] R. C Gonzalez, R. E Steven, et al. Digital Image Processing Using Matlab, Beijing: Publishing House Electronics Industry, 2004.

[11] C.Torrence, G. P.Compo, “A Practical Guide to Wavelet Analysis”, Bull. Am. Meteorolog. Soc., 1998, 79(1): 61-78.

[12] D. B.Percival, Walden, et al, Wavelet Methods for Time Series Analysis, Cambridge, U.K.: Cambridge University Press, 2000.

Extraction of Road Networks Using Matching Pattern and Mathematical Morphology Technology

Rong Liu, Rongcong Xu, Meiqing Wang
College of Mathematics and Computer Science, Fuzhou University
Fuzhou, Fujian, 35000
Email: liu_r@fzu.edu.cn, maths@fzu.edu.cn, mqwang@fzu.edu.cn

ABSTRACT

Road information is an important geomatic information in the study of remote sensing images. Its automatic extraction from remote sensing images will simplify classification and measures of objects in the images. But automatic extraction of road network is very difficult. In this paper we propose a semi-automatic extraction method. Through pattern matching and processing with mathematical morphology, a skeleton of road network can be obtained. Experimental results show that the method is quite accurate with simplicity and rapidity for middle or high-resolution remote sensing images.

Keywords: Mathematical Morphology, extraction of road network, image match.

1. INTRODUCTION

How to extract information for road networks in images of remote sensing is an important topic in geomatics research [1]. Road information extraction most relies on the resolution of remote sensing images, that is, extraction methods of road networks may vary with the spatial resolutions of the images. The ideal objective of road extraction is using an automatic system, which can get road network information with complete automation. But, the extraction process includes two stages: recognition and measure, the first stage is difficult to complete for computers which makes extraction of road networks with complete automation very difficult. So most research focus on semi-automatic system. In recent research, most processing methods of remote sensing images focus on edge extraction of the images [1-4]. The method proposed in paper [2] uses mathematical morphology to operate images, and get road networks using Douglas-Peuker algorithm. But this method also needs a good classified method for objects in images.

In this paper we proposed a semi-automatic extraction method for road network in remote sensing images. In the method, we first use matching pattern method to get an interesting part about road network, and then process the result image by using mathematical morphology to get a connected binary image for the trend of road network. Last, get skeletons of road network in images by the thinning method in [5]. In our experiments, we test a remote sensing image with 0.6m resolutions. The experimental results show that the method proposed in this paper is suitable for middle or high-resolution remote sensing images with simplicity and rapidity.

This work was partially supported by Fujian, China under Grant No.JB04036 and partially supported by Fuzhou University, China, under Grant No. 2005-XQ-16.

2. THE THEORY OF MATHEMATICAL MORPHOLOGY

Mathematical morphology [6], which started to develop in 1960s, stands as a relatively separate part of image analysis. It is based on the algebra of non-linear operators operating on object shape and in many respects supersedes the linear algebraic system of convolution. Morphological operations can be used for the image processing such as noise filtering, shape simplification, object structure enhancing, and image segmentation. Mathematical morphology is not only more quickly than other standard approaches but also has the natural structure for parallel computation. Some morphology operators used in this paper are described in detail [7].

The basic elements in Mathematical morphology are sets, which represent objects in images. The elements are still considered as (small) images. The morphological operators deal with two images. The image being processed is called as the active image, and the other image being a kernel is called as the structuring element. The primary morphological operators are erosion and dilation. Dilation can be used to fill small holes and narrow gulfs, and erosion can be used as noise filter and simplifying the structures of objects.

Let $\Re^n$ be n-dimensional Euclidean space. Suppose A and B be subsets of $\Re^n$.

The dilation of active image A by structuring element B is defined as following:

$$A \oplus B = \{c \in \Re^n \mid c = a + b \quad for \quad some \quad a \in A \quad and \quad b \in B\}$$

and the erosion of A by B is defined as:

$$A \Theta B = \{x \in \Re^n \mid x + b \in A \quad for \quad every \quad b \in B\}$$

Except dilation and erosion, two other important morphological transformations are opening operation and closing operation. The opening operation of an image A by the structuring element B is denoted by $A \circ B$ and is defined as

$$A \circ B = (A \Theta B) \oplus B$$

and the closing operation of an active image A by the structuring element B is denoted by $A \circ B$ and is defined as

$$A \bullet B = (A \oplus B) \Theta B$$

The operation used for thinning and thickening objects is the hit-or-miss transformation, which can be used for finding local patterns of pixels. It is defined as:

$A \otimes B = \{x : B_1 \subset A \text{ and } B_2 \subset A^C\}$ $B = (B_1, B_2)$,

A^C is the set complement of A.

Another operation we will use in this paper is the thickening transformation, which is defined as

$A \square B = A \cup (A \otimes B)$

3. THE ALGORITHM OF ROAD EXTRACTION

Consider the connectivity property of road network in remote sensing images, in this paper; we propose an algorithm of road extraction using pattern matching and mathematical morphology. The proposed algorithm is depicted as Fig 1.

3.1. Pattern Matching

The most common matching of image is pattern matching, called area-based matching or neighbor-based matching [8]. In this method, a template is set in the original image which road network needs to be extracted and is used as matched pattern. While matching is often based on directly comparing gray-level properties of image sub-regions, it can be equally well-performed using image-derived features or higher-level image descriptors, for example, the average of gray-level histograms. The more feature we known of matched target, the more perfectly pattern can be found. The matching algorithm is described as this, first shift the matched pattern over the origin image to compute the correlation between pattern and the searched image data, the criteria exceed the preset threshold represent pattern locations in the image.

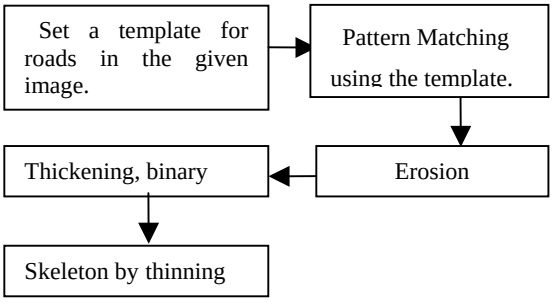


Fig. 1. stages of road extraction

matching criteria can be defined in many ways. Let *G* be an image to process, h be a pattern for which to match, and v be the set of image pixels in the pattern. In this paper, we proceed the matching by a common matching optimality criteria [6] as

$$C(u,v) = \frac{1}{(\sum_{(i,j)\in V}((G(u+i,v+j)-h(i,j))^2+1} \quad (1)$$

which stands for the matching value of pixel point *(u, v)* of the original image.

Considering the regular shape of roads, we select a small rectangle pattern from the road region of the given image. Let G be the remote sensing image of M×N pixels，H be a pattern of m×n pixels, and set a threshold as T. The following function constructs a set *F* of matching results.

$$f(u,v) = \begin{cases} 1 & if \quad C(u,v) \ge T \\ 0 & if \quad C(u,v) < T \end{cases} \quad (u,v) \in G \quad (2)$$

To avoiding extension of the boundary of *G*, let *f(u, v)*=0 if the point *(u, v)* inside the region sized m×n of right border and bottom border. So, we get the set F of result by shift through the *G*, *F={f(u, v)| (u, v)∈G }*. Now, we begin deal with the set F using mathematical morphology.

3.2. Application Of Mathematical Morphology

The set F is obtained after pattern matching, but F may contain the points of other objects which color is similar to the road. Consider F as a binary image, and the non-road information as the noise. So it is a denosing problem to filter that non-road information. In this stage mathematical morphological transformation erosion is used to filter noise.

In this paper, we use more than one 3×3 structuring elements.

Let $B = \{B_1, B_2, ...B_i, ..., B_{56}\}$ is the set of structuring elements, erosion $F \Theta B$ is defined by

$$F \Theta B = \bigcup_{i=1}^{56} (F \Theta B_{1i}) \quad (3)$$

The structuring element B_{1i} is composed of the three different arbitrary element, such as Fig 2. In order to avoid deleting some noise points, which may lead interruption of roads, some structuring elements of *B* are chosen to erode F.

Now, consider the thickening-structuring element different from erosion, composed of two 3×3 structuring elements. Let thickening structuring element $B = (B_1, B_2)$, B_1 is a 3×3 structuring element composed of the five different arbitrary pixel points, such as figure 3. B_2 is the complement of B_1, that is, $B_2 = B_1^c$. Let $B = \{B_i\}$, set thickening transformation $F \square B$ defined as

$$F \square B = \bigcup_{i=1}^{56} (F \square B_i) \quad (4)$$

To avoiding dilation too quickly, we also choose some of structuring elements from B

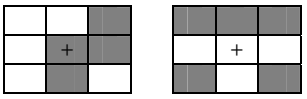


Fig. 2. structuring element of erosion
Fig. 3. structuring element of thickening

similar to the method in [9], to filter noises by erosion and dilation, we propose an algorithm by erosion and thickening to filter noise as following:

$$F = ((F \Theta B) \square B) \square B) \Theta B \quad (5)$$

After thickening iterating

$$F \square B_{\smile} = F \square \underbrace{B \square B \cdots \square B}_{k} \quad (6)$$

the road becomes connected.

Last, we get the binary image from the result image *F*.

3.3. Skeleton By Thinning

Skeleton is an important shape representation by simplifying region to graph. It can be obtained by thinning technology or mathematical morphology.

Median axis transform (MAT) is a thinning technology to get skeleton, defined as

$$S(X) = \{p \in X : \exists b_1, b_2 \in B \ \ b_1 \ne b_2 \ \& \ d(b_1, p) = d(b_2, p) = d_S(p, B)\}$$

$$d_S(p, B) = \inf\{d(p, b) \mid b \in B\} \quad (7)$$

B is the boundary of a region *X*。The distance metric that

is used depends on the grid and definition of connectivity, such as Euclidean distance, grid distance, etc. According to Eq. (7), the distance will be calculated between pixels of boundary and pixels inside of the region. [5] Proposed an applied method for getting skeleton of binary region quickly. The method for extracting the skeleton of a picture consists of removing all the contour points of the picture except those points that belong to the skeleton. In order to preserve the connectivity of the skeleton, it divides each iteration into two sub-iterations. In the first sub-iteration, the contour point P_1 is deleted from the digital pattern if it satisfies the following conditions:

$(a)\ 2 \le B(P_1) \le 6$

$(b)\ S(P_1) = 1$

$(c)\ P_2 \times P_4 \times P_6 = 0$

$(d)\ P_4 \times P_6 \times P_8 = 0$

Where $S(P_1)$ is the number of 01 pattern in the ordered set $P_2, P_3, P_4, \ldots P_8, P_9$ that are the eight neighbors of P_1 (Fig. 4), and $B(P_i)$ is the number of nonzero neighbors of P_1, that is,

$$B(P_1) = P_2 + P_3 + P_4 + \cdots + P_8 + P_9 = \sum_{i=1}^{9} P_i - 1$$

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Fig. 4. location of the nine pixels in a 3 x 3 window.

In the second sub-iteration, only conditions (c) and (d) are changed as follows:

$(c')\ P_2 \times P_4 \times P_8 = 0$

$(d')\ P_2 \times P_6 \times P_8 = 0$

and the rest remains the same.

4. EXPERIMENTAL RESULTS

In our experiments, a remote sensing image of 256*256 with 0.6m resolutions is tested. Fig 5 shows the original image and the results of each stage.

(a)

(b)

(c) (d) (e)

Fig. 5. the results of every stage
(a) the original image
(b) the result using matching pattern
(c) the result after filtering the noise of image (b)
(d) thickening image (c)
(e) Skeleton of (d)

5. CONCLUSION

In this paper, we propose a extraction method of road network using pattern matching and mathematical morphology. It can extract road information quite accurately and is suitable for high or middle resolution images. It is not good for the images with low resolution because it uses pattern matching which compares pixel values of pattern with that of the original image. During the study, we find that shadows of building will affect the road extraction. Our future work will focus on the inpainting of the shadows in roads. Another research area will focus on how to recognize the objects in images.

6. REFERENCES

[1] J.Ling, Y.S.Zhao, et al, "A Research for City Road Extraction from Satellite", Remote Sensing Technology And Application, 2005, 20(5): 478-482.

[2] R.An, X.Z.Fen, et al, "Road Feature Extraction from Remote Sensing Classified Imagery Based on Mathematical Morphology and Analysis of Road Networks", Journal of Image and Graphics, 2003, 8(7): 799-804.

[3] H.Lin, P.J.Du, et al, "Edge Detection Method of Remote Sensing Images Based on Mathematical Morphology of Multi-Structural Elements", Remote Sensing Technology And Application, 2004, 19(2).

[4] X.J.Zha, L.X.Xing, et al, "Method for Boundary Information Extraction from Remote Sensing", Journal of Jilin University (Information Science Edition), 2005, 23(2): 195-198.

[5] T.Y.Zhang, C.Y.Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns", Comm AMS, 1984,27: 236-239.

[6] M.Sonka, V.Hlavac, et al, Image Processing, Analysis, and Machine, Thomson Asia Pte Ltd.

[7] F.Y.Shih, S.X.Cheng, "Adaptive mathematical morphology for edge linking", Information Science, 2004, 167: 9-21.

[8] Y.J.Zhang, "Image Understanding and Computer Vision", Tsinghua Press, 2000, 163-186.

[9] Y.J.Zhang, "Image Processing and Analysis", Tsinghua Press, 2000, 256~269.

Design Philosophy for Patten Recognition Using Nano-devices*

Xuejun Chen，Guochen Ding，Mingxia Ji
Aviation Electronic Department, Naval Aeronautical Engineering Academy Qingdao Branch
Qingdao, Shandong 266041, China
Email: cxj0532@163.com*

ABSTRACT

We propose a novel approach for signal pattern recognition analysis using quantum dots (QD) based on the nano-scale devices subsume a profound understanding of the complex dynamics of small arrays of quantum structures, robust behavior of the arrays can be harnessed for unconventional. Our methodology combines an ultra fast neuromorphic learning algorithm in the QD array.

Keywords: Pattern recognition, Single electron tunneling, quantum dots, neural synapse.

1. INTRODUCTION

In this paper, in order to fully exploit exciting opportunities for revolutionary advances in *n*ano-scale computing, communication, detection, and sensing etc, we subsumes a profound understanding of the complex dynamics and properties of small arrays of quantum structures, such as QD, QD lasers, and others. Such arrays produce robust bi-stable and multi-stable behavior, which can be harnessed for unconventional, yet powerful computational concepts, and in particular for neuromorphic computing. Our aim is to realize nano-devices for improving capability of QD array on processing complex information, such as pattern recognition etc, and we demonstrate a capability for pattern classification using neuromorphic algorithms.

2. TRANSFER CHARACTERISTICS IN QD ARRAYS

Consider a one-dimensional array of N tunnel junctions constructed from metallic source and drain electrodes weakly coupled to a linear array of N - 1 metal clusters, see Fig. 1..

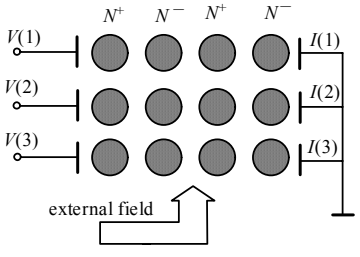


Fig.1 One-Dimension QD arrays and its Input V output I

We review the results of the "orthodox theory" of single-electron tunneling [1] to describe the charge transport through the array under small, but finite, bias voltage. The vector $\vec{n}$ defines the state of our system, $\vec{n}$ = (n_1,… n_i, …n_{N-1}), where n_i is the number of excess electrons accommodated by the *ith* QD. The Gibbs free energy E ($\vec{n}$, V) describing the electrostatic energy of the array of QD and its interaction with the external voltage is

$$E(\vec{n},V)=\frac{1}{2}\sum_{i=1}^{N-1}C_{ii}\phi_i^2+\frac{1}{2}\sum_{i=1}^{N}\sum_{j=0,j\neq 1}^{N-1}C_{ii}(\phi_i-\phi_j)^2-V_sQ_s-V_dQ_d\alpha \qquad (1)$$

where C_{ij}= C_{ji}, is the mutual capacitance between conductors i and j , and φ_i is the electrical potential of cluster i measured with respect to the substrate. The source and drain electrodes are enumerated as i = 0 and i=N, respectively. The source potential isφ_0=V_s= V / 2 , and the drain potential isφ_N=V_d=- V / 2. V is the transport voltage across the array. The charge on the source electrode is Q_s = $C_{01}(\varphi_0-\varphi_1)$ +en_s, and the charge on the drain electrode is Q_d = $C_{N-1}(\varphi_N-\varphi_{N-1})$+$en_d$, where n_s(n_d) is the number of electrons that has tunneled from the source (drain) electrode through the first (last) junction.

To determine the free energy, the potentials $\vec{\phi}=(\phi_1\cdots\phi_{N-1})$, must be determined from the static charge configuration. Using the charge conservation law, the total charge on island i can be written in terms of the potentials

$$q_i=C_{ii}\phi_i+\sum_{j=0,j\neq 1}^{N}C_{ii}(\phi_i-\phi_j)^2=en_i+q_{0,i},i=1,\cdots N-1 \qquad (2)$$

The background charges –e/2 <$q_{0,i}$< e/2 are due to incompletely screened charges in the environment of the islands. Equation (2) can be written in matrix form $\vec{Q}=C\vec{\phi}$, where the generalized capacitance matrix elements are defined

$$C_{ii}=\sum_{j=0}^{N}C_{ij};\qquad C_{ii}=-C_{ij} \qquad (3)$$

and the augmented charge vector is defined as Q_i≡q_i+ C_{i0}+ $C_{iN}\varphi_N$.

The generalized capacitance matrix can be inverted to obtain the potential distribution given any charge distribution. For convenience, we rewrite the free energy of the array using matrix notation as

$$E(\vec{n},V)=\frac{1}{2}\vec{Q}^TC^{-1}\vec{Q}_s-V_sQ_s-V_dQ_d \qquad (4)$$

In describing the electron transport through the array, we neglect here the effects of co-tunneling[①], and consider only single-electron tunneling between nearest neighbors in the array. That is, the final state $\vec{m}$ of the tunneling differs from the initial state $\vec{n}$ by the transfer of a single electron though the k^{th} junction, e.g. $\vec{m}=\vec{n}\pm\Delta\hat{u}_k$, where $\Delta\hat{u}_k\equiv\hat{u}_k-\hat{u}_{k-1}$ and $\hat{u}_k$ is a unit vector for the k^{th} QD. The ± sign gives the direction of tunneling through the junction. If the transition rates Γ are sufficiently small, one can perform a calculation using Fermi's Golden Rule to obtain [1]

$$\Gamma_k^{\pm}(\vec{n},V)=\frac{1}{2}\Gamma_k^{\pm}\left[\Delta E_k^{\pm}(\vec{n},V)\right]=\frac{-\Delta E_k^{\pm}}{e^2R_k}\left[1-\exp\left(\frac{\Delta E_k^{\pm}}{k_BT}\right)\right]^{-1} \qquad (5)$$

where $\Delta E_k^{\pm}(\vec{n},V)\equiv E(\vec{n}\pm\Delta\hat{u}_k,V)-E(\vec{n},V)$ is the change in the

*The authors wish to acknowledge each other due to the many helpful discussions among with them.

free energy of the system due to the tunneling, R_k is the effective resistance of the tunnel junction, $e > 0$ is the fundamental unit of charge, and the thermal energy is k_BT. One can use probability conservation to write the corresponding master equation describing the time evolution of the probability $P(\vec{n},t)$ of finding the circuit in the state $\vec{n}$

$$\frac{dP(\vec{n},t)}{dt} = \sum_k [\Gamma_{k-1}^+(\vec{n}-\Delta\hat{u}_{k-1})P(\vec{n}-\Delta\hat{u}_{k-1},t) + \Gamma_k^-(\vec{n}+\Delta\hat{u}_k)\cdot P(\vec{n}+\Delta\hat{u}_k,t)] - \sum_{k-1}^{N}\left[\Gamma_k^+(\vec{n}) + \Gamma_k^-(\vec{n})P(\vec{n},t)\right] \quad (6)$$

Practical approaches to solving the master equation are described in [3,4], the average tunneling current is given by computing the net flow through any junction k in the array

$$I(V) = I_k(V) = e\sum_{\vec{n}} P(\vec{n})\left[\Gamma_k^+(\vec{n},V) - \Gamma_k^-(\vec{n},V)\right] \quad (7)$$

Since the summation is performed over the charge states, the current is a function of the transport voltage. Fig. 2. shows the *I*—*V* characteristic of QD arrays.

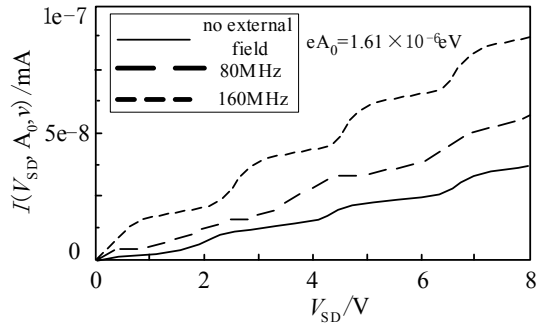


Fig.2 *I*—*V* characteristics of One-Dimension arrays

3. NEUROMOPHIC ALGORITHMS IN QD ARRAYS

QD nano-electronic devices represent a promising hardware technology that offers both conceptual opportunities and engineering challenges for complex information processing applications [2, 3]. One such application, namely pattern recognition, is of considerable interest to the development of modern intelligent systems and will be considered here. In recent years, the quest for innovative approaches to machine intelligence has received considerable attention. The proven ability of neuromorphic algorithms to deal with uncertain information and to interact with dynamic environments is therefore providing a strong incentive to explore the feasibility of their implementation on arrays of QD. However, in contrast to conventional hardware approaches, we must develop here computational paradigms that exploit from the onset not only the concept of massive parallelism but also, and most importantly, the physics of the underlying device.

Artificial neural networks are adaptive systems that process information by means of their response to discrete or continuous input [4]. Neural networks can provide practical solutions to a variety of artificial intelligence problems, including pattern recognition [5], autonomous knowledge acquisition from observations of correlated activities [6], realtime control of complex systems [7], and fast adaptive optimization [8]. At the heart of such advances lies the development of efficient computational methodologies for "learning". The development of neural learning algorithms has generally been based upon the minimization of an energy-like neuromorphic error function or functional. Gradient-based techniques have typically provided the main computational mechanism for carrying out the minimization process, often resulting in excessive training times for the large-scale networks needed to address real-life applications. Consequently, to date, considerable efforts have been devoted to: (1) speeding up the rate of convergence and (2) designing more efficient methodologies for deriving the gradients of these functions or functional with respect to the parameters of the network. The primary focus of such efforts has been on recurrent architectures. However, the use of gradient methods presents challenges even for the less demanding multi-layer feed-forward architectures, which naturally occur in QD arrays. For instance, entrapment in local minima has remained one of the fundamental limitations of most currently available learning paradigms. The recent development of the innovative global optimization algorithm has been suggested as a promising new avenue for addressing such difficulties.

We illustrated the underlying physical concepts of single-electron transport in arrays of QD there is a profound similarity between the dynamics of neural networks and that of QD arrays. In the latter, the free energy of an array characterized by a charge distribution can be lowered in terms of tunneling events. For neural networks, on the other hand, Hopfield has shown that the stable states of the network are the local minima of a bounded Lyapunov function of the net's output parameterized by the synaptic interconnection weights. A careful analysis, however, reveals that this formal similarity is not adequate for implementing learning algorithms for pattern recognition. By comparing the leading terms of the free energy in (4), i.e., $1/2\vec{Q}^TC^{-1}\vec{Q}$ and the Lyapunov function in a Hopfield network, i.e., $L(x,W) = -1/2x^TWx$, we see that the inverse of the augmented capacitance matrix would have to play the role of the synaptic matrix. However, the elements of C_{ij} are fixed, and cannot be modified. An alternative approach for controlling the dynamics of the system has to be found. In principle, one could manipulate the free energy of the array via capacitive gating of each of the QD.

In the absence of a time-dependent field, current flows through a QD via tunneling when an unoccupied internal energy state is aligned to the Fermi energy of the leads. However, if a time-varying AC voltage $A_0\cos(2\pi\omega t)$ is applied, inelastic tunnel events are induced when electrons exchange photons of energy $\hbar v$ with the oscillating field. A direct inclusion of this phenomena in a master equation that takes into account Coulomb blockade can be made by writing the tunneling rate $\overline{\Gamma}$ through each barrier in the presence of an electromagnetic excitation in terms of the rates without the external AC field

$$\overline{\Gamma}_k^{\pm}(\vec{n},V,A_0,v) = \sum_{\alpha=-\infty}^{\infty} J_\alpha\left(\frac{eA_0}{\hbar v}\right)\Gamma_k^{\pm}\left(\Delta E_k^{\pm} + \alpha\hbar v\right) \quad (8)$$

Where J_a denotes the Bessel function of the first kind and a denoting the number of photons exchanged. This generalized master equation is obtained by substituting the rates in (8) into (6). The current through this device becomes a function of the transport voltage V, and the amplitude A_0 and frequency v of the AC field

$$I(V,A_0,v) = e\sum_{\vec{n}} P(\vec{n})\left[\overline{\Gamma}_k^+(\vec{n},V,A_0,v) - \overline{\Gamma}_k^-(\vec{n},V,A_0,v)\right] \quad (9)$$

We consider the transport voltage V as the input variable and the current I as the output function in designing the neuromorphic computation. For a two-dimensional QD array with K input and K output nodes, we can readily generalize the description given here to consider K input voltages V_k and K output currents $I_k(V_k,A_0,v)$. This vector valued function $\mathbf{I}_K$ is controllable through the parameters of the external, alternating field, A_0 and v, by minimizing the error function E, defined over the number of L training

patterns as the squared difference between the *l*-th observed current, ${}^{l}\mathbf{I}_K$ and the target currents, ${}^{l}\mathbf{I}^*_K$ (see below)

$$E \equiv \frac{1}{L}\sum_{l=1}^{L}\left[{}^{l}\mathbf{I}_K(\mathbf{V}_K, A_0, v) - {}^{l}\mathbf{I}^*_K\right]^2 \qquad (10)$$

Matrix and vector dimensions are explicitly indicated as subscripts. If a larger number of controls are necessary, then a polychromatic AC field may be considered for the global control, rather than a monochromatic field.

4. PATTERN RECGNITION USING QD ARRAYS

The pattern recognition scheme can now readily be implemented using the following method. We will assume that two sets of L vectors are used for training. They are stored as rows of two matrices $\mathbf{\Omega}_{LI}$ and $\mathbf{R}_{LO}$ respectively, which represent the input signal patterns and the target outputs. We denote the number of columns of each matrix as I for input and O for output, without confusion. Since typically $L >> I$, two preprocessing steps are used. First, appropriate clustering transforms $\mathbf{\Omega}_{LI}$ and $\mathbf{R}_{LO}$ into smaller size matrices $\mathbf{\Omega}^*_{KI}$ and $\mathbf{R}^*_{KO}$. Then, two successive nonlinear transformations map $\mathbf{\Omega}^*_{KI}$ into $\mathbf{H}_{KK}$, a nonsingular K x K presynaptic matrix, which constitutes the actual input into the quantum-dot array. We also decouple the nonlinearity of the transfer function, φ, at the output layer of the neural net from the linear interlayer pattern propagation mediated by the synaptic weights $\mathbf{W}_{KO}$. This transformation nominally computes the postsynaptic input to the output layer of the neural net as a K x O rectangular matrix.

Since the latter is connected via a bijective sigmoid mapping to the output training examples, the synaptic interconnection matrices $\mathbf{W}_{KO}$ can be determined by solving the linear system $\mathbf{H}_{KK}\ \mathbf{W}_{KO} = \varphi^{-1}(\mathbf{R}^*_{KO})$ using gradient iteration. In simulation on a conventional computer, this can be accomplished by exactly solving the system of linear equations using singular-value decomposition techniques. On nano-electronic hardware, the same result can be achieved by directly optimizing the error function in (10), where the sum is now running from 1 to K clusters, in terms of the parameters of the external field, A_0 and v. If the dimension O of the output pattern is smaller than the number K of output nodes of the QD array, the output error is calculated using the Euclidean distance in O dimensions.

5. DIFFICULTIES AND FUTURE WORKS

Difficulties in our works will focus on directly accounting for uncertainties when attempting to obtain best estimates for the device parameters and responses of interest. For example, nominal values for the elements of the capacitance matrix will be computed from "first-principles" simulations of the metal clusters and substrate via density functional-theory-based molecular dynamics [1] and the current through the device will be computed via numerical solutions to the master equation. To obtain best estimates for critical parameters (e.g., A_0 and v), we must consistently combine computational results and experimental measurements.

The next work is to optimize a generalized Bayesian loss function that simultaneously minimizes the differences between the best estimate responses and the measured responses on one hand, and the best estimate and calculated parameters on the other hand, and the optimization process will use the inverse of a generalized total covariance matrix as the natural metric of the calculational manifold in conjunction with response sensitivities to all parameters.

6. REFERENCES

[1] K.K. LIKHAREV, "Single Electron Devices and their Applications", Procs. IEEE, 1999，87(4) : 606-632.
[2] Chen RH, K.K. Likharev, "Single electron transistor logic," Appl.Phys. Lett. 1996, 68(14): 1954-1956.
[3] L.R.C. Fonseca, K.K. Likharev, et al, "A Numerical Study of the Dynamics and Statistics of Single-Electron Systems", J. Appl. Phys. 1995, 78: 32-38.
[4] M. Hassoun, Fundamentals of Artificial Neural Networks, MIT Press, 1995.
[5] M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1997.
[6] M. Beckerman, Adaptive Cooperative Systems, Wiley Interscience, 1997.
[7] D. White, D. Sofge, Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches, Van Nostrand Reinhold, 1992.
[8] J. Goossens, A.H.M. van Roemund, "Learning Single Electron Tunneling Neural Nets," Proceedings of the ProRISC Workshop on Circuits, Systems and Signal Processing 1997, 179-186.

A SOLUTION TO THE REPRODUCING KERNELS SPACE

Qinli Zhang1,2
1School of Information, Southern Yangtze University, Wuxi Jiangsu, 214122,China;
2North China Institute of Aerospace Engineering, Langfang Hebei, 065000, China
Email: zhangql1972@sina.com
Shitong Wang
School of Information, Southern Yangtze University, Wuxi Jiangsu, 214122, China;
Email: wxwangst@yahoo.com.cn

ABSTRACT

In this paper, a new but simple solution method to the reproducing kernels of the reproducing kernel space is proposed. The proposed method is based on the use of both the reproducing kernel functions and the convolution operator.

Keywords: reproducing kernel; radial basis function ; support vector machine; Basic solution; Fourier transform

1 INTRODUCTION

For the sake of the reproducing kernel function with many good properties, it had been widely applied to many fields (see [1]-[3]). Accordingly, many researchers show great interest in the reproducing kernel and pay much attention to it. But it was very difficult to calculate the reproducing kernel function. At present, there is only a method of calculating the reproducing kernel of $H^n(R)$, namely, by solving the differential equations including δ function (see [4]). When n becomes very large, the method will not work.. In this paper, we successfully solved the problem by the use of $K_1(x)$ and the convolution operator. Whenever n is small or large, the new method makes the corresponding complicated and inconvenient calculation become very easy. This is the advantage of this proposed method here.

2 PRELIMINARIES

Definition 1 Let X be an abstract set, and H be a Hilbert space of real or complex value functions f defined on the set X . A function $K(x,y)$ on $X\times X$ is called a reproducing kernel, if $K(x,y)$ satisfies the following two properties:

(1) $K(x,\cdot)$ belongs to H for all $x\in X$;

(2) $<f(y),K(x,y)>_y = f(x)$

for all $x\in X$ and all $f\in H$.

In this paper, we denote $H^n(R)$ as the set of real functions that are absolutely continuous and square-integrable, with absolutely continuous derivatives up to the order $n-1$ and square-integrable derivatives up to the order n on R , where n is a positive integer. We denote δ as the Dirac function, F as the Fourier transformation and F^{-1} as the Fourier inverse transformation.

To compute reproducing kernels , we usually try to solve a boundary value problem directly for a differential equation, or to make use of some properties of reproducing kernels. For example, if H is a separable Hilbert function space having a reproducing kernel, and we know the orthogonal basis $\{\varphi_i(x)\}$ of H , then the reproducing kernel of H is simply obtained from $K(x,y)=\sum_{i=1}^{\infty}\varphi_i(x)\overline{\varphi}_i(y)$. Otherwise, we should solve a boundary value problem directly for a differential equation. However, it is not always easy to do so. We introduced a technique for computing the reproducing kernels of the two spaces $H^1(R)$ and $H^2(R)$ by the use of δ function.

Theorem 2.1 Let $K_1(x)$ be the basic solution of the operator $L=1-\frac{d^2}{dx^2}$, then the reproducing kernel of $H^1(R)$ associated with the inner product

$$<u,v>_{H^1(R)}=\int_R (uv+u'v')\,\mathrm{d}x$$
$$(u(x),v(x)\in H^1(R))$$

is $K_1(x-y)$.

Proof : According to the known condition of theorem, we have

$$L(K_1(x))=\delta(x),$$
$$-K_1''+K_1=\delta(x),$$

By the Fourier transform, we have

$$F(-K_1'')+F(K_1)=F(\delta(x)),$$
$$-(i\omega)^2F(K_1)+F(K_1)=1,$$
$$F(K_1)=\frac{1}{1+\omega^2} \qquad (1)$$

Therefore, by the Fourier inverse transform, we get

$$K_1(x)=F^{-1}(\frac{1}{1+\omega^2})=\frac{1}{2}e^{-|x|}.$$

By [5], we have that $K_1(x-y)$ is the reproducing kernel of $H^1(R)$.

Theorem 2.2 Let $K_2(x)$ be the basic solution of the operator $L=\frac{d^4}{dx^4}-2\frac{d^2}{dx^2}+1$, then the reproducing kernel of $H^2(R)$ associated the inner product

$$< u,v >_{H^2(R)} = \int_R (uv + 2u'v' + u''v'')\,\mathrm{d}\,x$$

$$(u(x), v(x) \in H^2(R))$$

is $K_2(x-y)$.

Proof : According to the known condition of theorem, we have

$$L(K_2(x) = \delta(x),$$

$$K_2^{(4)} - 2K_2'' + K_2 = \delta(x),$$

By the Fourier transform, we have

$$F(K_2^{(4)}) - 2F(K_2'') + F(K_2) = F(\delta(x)),$$

$$(i\omega)^4 F(K_2) - 2(i\omega)^2 F(K_2) + F(K_2) = 1,$$

$$F(K_2) = \frac{1}{(1+\omega^2)^2}.$$

Therefore, by the Fourier inverse transform, we get

$$K_2(x) = F^{-1}(\frac{1}{(1+\omega^2)^2}) = \frac{1}{4}(1+|x|)e^{-|x|}.$$

By [5], it follows that $K_2(x-y)$ is the reproducing kernel of $H^2(R)$.

3 MAIN RESULTS

In this part, we consider the Hilbert space $H^n(R)$ associated with the inner product

$$< u,v >_{H^n(R)} = \int_R \sum_{i=0}^{n} C_i u^{(i)} v^{(i)}\,\mathrm{d}\,x$$

$$(u(x), v(x) \in H^n(R)),$$

where $C_i (i = 0,1,2,\cdots,n)$ is the coefficient of expansion of $(a+b)^n$.

Theorem If

$$K_n(x) = (K_1 * K_{n-1})(x) = \cdots =$$

$$\underbrace{(K_1 * K_1 * \cdots * K_1)}_{n\uparrow}(x) \qquad (2)$$

then, $K_n(x-y)$ is the reproducing kernel of $H^n(R)$,namely,

$$< f(x), K_n(x-y) >_{H^n(R)} =$$

$$< K_n(x-y), f(x) >_{H^n(R)} = f(y)$$

$$(\forall f(x) \in H^n(R))$$

Example Consider $H^2(R)$ associated with an inner product as follows:

$$< u,v >= \int_R (uv + 2u'v' + u''v'')\,\mathrm{d}\,x$$

In terms of Theorem3.2, we have

$$K_2(x) = (K_1 * K_1)(x) = \frac{1}{4}\int_R e^{-|y|-|x-y|}\,\mathrm{d}\,y$$

$$= \frac{1}{4}(1+|x|)e^{-|x|}$$

Obviously, the result is in accordance with that of Theorem2.2.

4 THE PROPERTIES

Property 1. Assume $K_n(x,y)$ denote the reproducing kernel of $H^n(R)$, then

$$K_n(x,y) = K_n(x-y) = K_n(y,x).$$

Namely, it is a convolutional kernel and symmetric function.

Proof : The result is obvious by $K_n(x-y)$.

Property 2. $K_n(x)$ is fading quickly with the increasing n.

Property 3. Assume $i = 0,1,2,\cdots,2n-2$ respectively, $K_n^{(i)}(x)$ is a even function or odd function in turn, where n is a positive integer.

Proof : In terms of (1) and (3), we have

$$F[K_n(x)] = \frac{1}{(1+\omega^2)^n}.$$

Therefore, $K_n^{(i)}(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \frac{(i\omega)^i}{(1+\omega^2)^n} e^{ix\omega}\,\mathrm{d}\,\omega$.

According to the substitution of varible and the property of integral, we easily get $K_n^{(i)}(-x) = (-1)^i K_n^{(i)}(x)$ where

$i = 0,1,2,\cdots,2n-2$.

Property 4. Assume $i = 0,1,2,\cdots,2n-2$ respectively, $K_n^{(i)}(x)$ has odd order or even order vanishing moments in turn.

Proof: In terms of property 3, this conclusion is obvious.

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented an efficient but simple method to compute the reproducing kernel $K_n(x)$ of the reproducing kernel Hilbert space $H^n(R)$ associated with the inner product

$$< u,v >_{H^n(R)} = \int_R \sum_{i=0}^{n} C_i u^{(i)} v^{(i)}\,\mathrm{d}\,x$$

$$(u(x), v(x) \in H^n(R)),$$

where $C_i (i = 0,1,2,\cdots,n)$ is the coefficient of the expansion of $(a+b)^n$. We have also derived several promising properties of $K_n(x)$, including quick fading, vanishing moment, symmetry, regularity, even or odd function, wavelet function, and so on.

In this way, the corresponding computational complexity is greatly reduced without integrating. We had successfully applied $K_n(x)$ to solving partial differential equation. The results of simulation experiments show the feasibility and effectiveness s(see[5]). At present, the radial basis functions, wavelet networks, support vector machines, fuzzy logic systems and evolutionary computing are very hot topics. [see [7]-[8]). $K_n(x)$ is remarkably analogous with the radial basis functions, kernel functions and wavelets. We

are exploring to substitute $K_n(x)$ for the radial basis functions, kernel functions and wavelets. We believe that $K_n(x)$ is very attractive so that it can be effectively used to solve the kernel-based problems. Applying $K_n(x)$ to the other fields remains to be explored in future.

REFERENCE

[1] Zhang Mian & Cui Minggen & Deng zhongxing. A new uniform convergent iterative method by interpolation where error decreases monotonically, J.Comput Math., pp.365-372,Mar., 1985.

[2] Rodriguez. G. & seatzu, S. Numerical solution of the finite moment problem in a reproducing kernel Hilbert space. J.Comp, Appl.Math., pp. 233-244,Nov.,1990.

[3] Xie Xianggen & Nashed, M.Z. The Backus-Gilbert methods for signals in reproducing kernel Hilbert spaces and wavelet subspaces., Inverse Problems. pp.785-804, Oct.,1994.

[4] Cui Minggen. Wavelet analysis of differential operator spline in $H^1(R)$ space. Ukr Acad. Sci Dokl. Math., pp.1221-1232,Oct.,1996.

[5] Zhang Qinli.The numerical methods for solving Euler system of equations in reproducing kernel space $H^2(R)$ .Journal of Computational Mathematicspp.327-336,Mar.,2001..

[6] W.Z. Pan, Fourier analysis and its application. Beijing: Peking University Press, 2000.

[7] Qinghua Zhang and Albert Benveniste, Fellow, IEEE. Wavelet Networks. IEEE Transaction on Neural Networks. pp.889-898,Mar.,1992.

[8] Jun Zhang, Gilbert G.Walter, etc. Wavelet Neural Networks for Function Learning. IEEE TRANSACTION ON SIGNAL PROCESSING, pp.1485-1497,June,1995.

Urban Ecosystem Early-Warning Based on Artificial Immune

Huimin Wang, Junfei Chen, Mingfeng Wu
State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering, Hohai University
Business school, Hohai University
Nanjing, Jingsu 210098, China
Email：huiminwang63@hotmail.com

ABSTRACT

Urbanization impels social advancement and breaks the balance of human society and environment simultaneously. The sustainable development of urban ecosystem is threatened. In this paper, Urban Ecosystem Early-Warning Based on Artificial Immune is studied. The idea of natural immune system is used for reference and immune mechanism of urban ecosystem is analyzed. The advanced model of urban ecosystem early-warning based on artificial immune is developed. Nanjing urban ecosystem as a case is studied for its sustainable development. The Early-Warning Indexes System is established and the results of Early-Warning show that the model is effective for monitoring and forecasting the sustainable development of urban ecosystem. It makes dynamic monitoring and forecasting on sustainable development of urban ecosystem instead of static evaluation.

Keywords: Artificial Immune, Urban Ecosystem, Early-Warning, Indexes System, Sustainable Development

1. INTRODUCTION

Along with the economy development of China, the cities are facing the unprecedented changes in recent years. The scale of city unceasingly expands, the massive population gathers to the city by unprecedented speed and the quantity of cities increase rapidly. On the other hand, the status of the city as the political, economic, cultural and social life main carrier is day by day prominent. The urbanization advances rapidly, at the same time, a series problems such as environmental pollution, ecology unbalance and social problems challenge the sustainable development of the city critically.

Since the 1980s, some researches about the urban ecosystem have been done and some results are obtained. In 1980, the second European symposium on ecology, which took the urban ecosystem as the central subject of the conference and explored from the aspects of theory, method, practice, application and so on. Forester, Vester and Hester studied the development tendency of urban ecosystem [1]. Odum thought the urban ecosystem and the natural ecosystem have the similar succession rule, which all have the process of occur, development, prosperous, undulation, decline and so on, and the process of city succession is the process of the energy unceasingly gather[2]. Researches about urban ecosystem and urban sustainable development started at the beginning in 1980s in China. The famous ecologist Ma and Wang explicitly pointed out that city is a typical "society - economy – nature" compound ecosystem [3]. Gu proposed the direction of sustainable development research of Chinese cities with all levels [4]. Xu and Hou did some quantitative research about the urban sustainable development using the principal components analytic method [5]. Pu and Xu studied the index system and model of environmental capacity for urban sustainable development [6].

At present, there are most static evaluations about urban ecosystem in the literature. However, the evaluation of urban ecosystem sustainable development is a process of static evaluation combining with dynamic monitor. Therefore, it is very important to research and monitor the ecology environment and ecology security, then forecast the unusual condition which will possibly appear in the future. In this paper, the model of urban ecosystem early-warning based on the natural immunity is established, which is one of the important way to prevent the urban ecosystem from disorder development, and it will play a great significance for monitoring the condition of urban ecosystem, promoting the sustainable development of urban ecosystem and improving human survival environment.

2. MODEL OF URBAN ECOSYSTEM EARLY-WARNING BASED ARTIFICIAL IMMUNE

2.1 Feasibility Analysis of Urban Ecosystem Immune Early-Warning

The natural immune system is a complex auto-adapted system, which is a complex disperse system composed of immune organs, organizations and cells. It can distinguish itself from others and has the function of removing and memorizing others. From the viewpoint of information processing, the interior immune system contains the information distribution, storage, recognition, study, memory and so on, which is a complex information processing mechanism. This system mainly displayed three kinds of functions: immune defense, immune self-stabilization and immune monitor[7].

The early-warning of urban ecosystem takes nature, society and economy coordination as a core, including the stability of natural ecology environment system, the fairness of social distribution and the sustainability of economy development. The immune mechanism of the natural immune system provides some enlightenment to the early-warning of urban ecosystem. The urban ecosystem, which is a sustainable development system, doesn't pursue the sole achievements of some one system, but the optimal of the whole integrated function: ecology security, economy prosperity and society fairness.

In a word, the natural immune system and the city ecology early-warning have the similar function, so the model of urban ecology early-warning can be established by immune method and used to urban ecology monitor, defense and early-warning.

2.2 Model of Urban Ecosystem Early-Warning

Both Dasgupta and Forrest once proposed negative algorithm according to the natural immune system and applied it in pattern recognition, fault diagnosis and data detection [8, 9]. The researchers applied this algorithm into Mackey Glass sequence, through examining the appropriate test data and some simulation data, they found that this algorithm is effective for examining the deviation of any normal pattern and don't need the prior knowledge of abnormity aspect. It provides a new thinking for establishing the model of urban ecosystem early-warning. However, for the algorithm is expressed with a group of binary data, which often used for fault diagnosis, pattern recognition and so on, it can't be applied to urban ecosystem directly and need to be improved for adapting urban ecosystem early-warning.

In the urban ecosystem early-warning model, Data for early-warning testing and match is produced randomly according to even distribution, which can ensure early-warning testing symmetrical and diversity in early-warning space and can reduce the number of invalid data. The theory of shape-space can ensure limited early-warning tested data overlaying the early-warning space. In addition, Euclidean distance between early-warning tested data and no warning data are calculated to complete "immunological tolerance". Then, early-warning tested data is optimized, reduced and clustered, which produce early-warning pattern aggregate. At last, urban ecosystem is inspected and early-warning by early-warning patterns. Figure 1 shows the process of immune early-warning algorithm, which include the following mainly steps.

1). **Data Encoding.** In order to produce early-warning testing aggregate R, data is produced in every warning degree according to random even distribution, and is expressed by decimal number. The scale of space V_ε, where V_ε is a recognition region, threshold value ε and the testing number of n should be evaluated properly, and it can make the limited data overlay the whole early-warning space. The maximum and minimum value of every index should be found, and all indexes data are unitary between 0 and 1, all these values will be looked as early-warning samples.

2). Affinity Function. Affinity between antibody and antigen is based on the comparability of their structure. Antibody or antigen is expressed by a real number coordinate aggregate $M = (m_1, m_2, \cdots, m_L)$, which can be looked as a dot in L dimension real number space $M \in S^L \subseteq R^L$, S is the shape-space, L is the dimension. If the coordinate of a antibody is defined as $(ab_1, ab_2, \cdots, ab_L)$ and the coordinate of antigen is defined as $(ag_1, ag_2, \cdots, ag_L)$, the affinity between antibody and antigen can be defined as the distance between these two coordinates, such as Euclidean distance, Manhattan distance and Hamming distance.

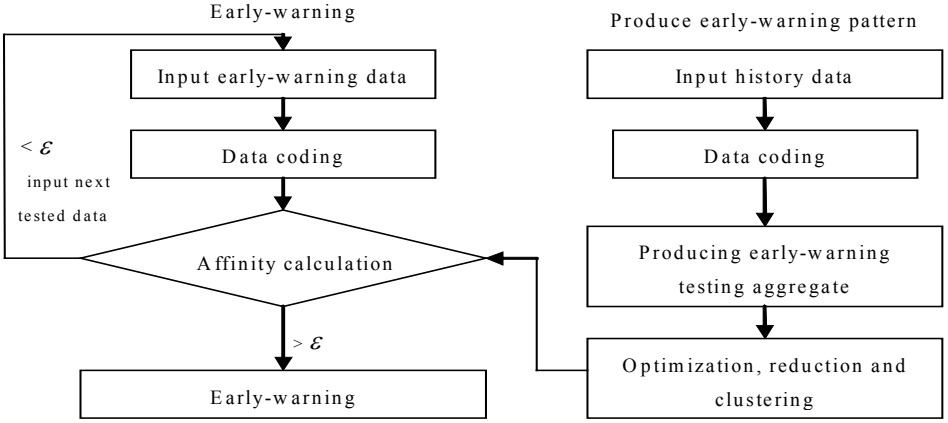


Fig. 1. Flow of the model of immune early-warning algorithm

Affinity function is defined as a distance function in the early-warning model, which is defined as follows:

$$D = (ab - ag)^2 \quad (1)$$

Where ab refers to early-warning tested data, ag refers to sample data of different warning degree, D refers to the distance between antibody and antigen. The smaller the distance function value is, the larger the affinity is.

3). Creating Early-Warning Testing Aggregate R. The early-warning testing aggregate R can be produced through randomly. The data should be cancelled if the affinity of testing aggregate data and the no warning aggregate data is smaller than ε. Repeat this process until all aggregates which demand testing are tested. The number n of testing aggregates is decided according to the practical problem.

4). Reducing Early-Warning Testing Aggregate. Affinity is calculated between the data in R and sample data of every warning data. Early-warning pattern aggregate R' is formed by reducing and clustering R according to the calculated affinity. The result of clustering is the mapping of every warning degree, and every early-warning pattern subset reflects the prominent character of one warning degree, which is sensitive to the warning data.

5). Early-Warning and Forecasting. The same

method is used to handle practical data when forecasting urban ecosystem, namely the same method of data encoding is adopted. Then calculating the affinity between practical data and R', and forecasting the whole ecosystem.

Table 1. Indexes of Nanjing Ecosystem Early-warning System

Target Layer	Standard Layer	Index Layer (weight)	Huge Warning	Heavy Warning	Middle Warning	Light Warning	Non-warning
A: Early-warning index sign of urban ecosystem sustainable development	B1:Early-warning index of the economic sub-system (0.3422)	C1:Per Capita GDP (0.0701)	<5000	[5000, 15000)	[15000, 50000)	[50000, 100000)	≥ 100000
		C2:Per Capita Sate Revenue (0.0674)	<2000	[2000, 5000)	[5000, 10000)	[10000, 15000)	≥ 15000
		C3: Growth Rate of GDP (0.0328)	<4	[4,7.2)	[7.2,15)	[15,30)	≥ 30
		C4: Percentage of Whole Social Fixed Assets Investments Occupied GDP (0.0227)	<20	[20,30)	[30,45)	[45,55)	≥ 55
		C5: Production Rate of Industry Whole Member Labor (0.05)	<2	[2,6)	[6,12)	[12,15)	≥ 15
		C6: Comprehensive Energy Consume of Per 10000 Yuan GDP (0.054)	>2	(1.3,2]	(0.8,1.3]	(0.5,0.8]	≤ 0.5
		C7:Percentage of Tertiary Industry's Incensement in GDP (0.0296)	<10	[10,28)	[28,40)	[40,50)	≥ 50
		C8:Rate of Total Amount of Foreign Capital Actually used to GDP(0.0303)	<3.0	[3.0,7.0]	[7.0,15)	[15.0,20.0)	≥ 20.0
	B2:Early-warning index of the social sub-system (0.3065)	C9: Unemployment Rate (0.0214)	>5	(3.5,5]	(2.5,3.5]	(2,2.5]	≤ 2
		C10: Engel Coefficient (0.0362)	>50	(40,50]	(30,40]	(25,30]	≤ 25
		C11: Population Natural Growth Rate (0.0844)	>30	(10,30]	(4,10]	(1,4]	≤ 1
		C12:Population Average Prospective Life Span(0.034)	<60	[60,70)	[70,78)	[78,82)	≥ 82
		C13:Proportion of Non-agriculture Population in City Population (0.0246)	<10	[10,22)	[22,60)	[60,90)	≥ 90
		C14:Per Capita Floor Space of Houses(0.0428)	<5	[5,12)	[12,20)	[20,25)	≥ 25
		C15:Percentage of Scientific and Technological, Educational Expenditure to GDP (0.0109)	<1.0	[1.0,1.7)	[1.7,2.5)	[2.5,3.0)	≥ 3.0
		C16:Number of Doctors Per 10000 Population(0.0194)	<20	[20,45)	[45,70)	[70,100)	≥ 100
		C17:Number of Books Stored in Public Library Per 100 Person(0.0131)	<100	[100,200)	[200,350)	[350,400)	≥ 400
	B3:Early-warning index of the natural sub-system (0.3532)	C18:Green Coverage Rate of City (0.0299)	<10	[10,25)	[25,40)	[40,60)	≥ 60
		C19:Per Capita Public Green Area Urban Households (0.0328)	<2	[2,8)	[8,15)	[15,25)	≥ 25
		C20: Per Year Density of TSP (0.0682)	>0.3	(0.2,0.3]	(0.13,0.2]	(0.08,0.13]	≤ 0.08
		C21:Average Value of Regional Environmental Noise (0.017)	>65	(57,65]	(50,57]	(45,50]	≤ 45
		C22:Rate of Drinking Water Resources of Reaching Water's Quality Standard(0.0617)	<85	[85,90)	[90,96)	[96,99)	≥ 99
		C23:Rate of Processed Industrial Waste Water of Reaching Standard (0.0246)	<40	[40,60)	[60,80)	[80,90)	≥ 90
		C24:Dirty Processing Rate of Water of City(0.0591)	<30	[30,55)	[55,80)	[80,90)	≥ 90
		C25: Treatment Rate of Industrial Solid Wastes (0.0169)	<80	[80,85)	[85,90)	[90,95)	≥ 95
		C26:Treatment Rate of Industrial Waste Gas(0.053)	<30	[30,45)	[45,60)	[60,70)	≥ 70

Table 2. Indexes Data Forecasting Value and Early Warning Result

Index	2005 Data	Warning Situation	Index	2005 Data	Warning Situation
Per Capita GDP	29407.14	Middle Warning	**Per Capita State Revenue**	4725.75	Heavy Warning
Growth Rate of GDP	13.06	Middle Warning	**Percentage of Whole Social Fixed Assets Investments Occupied GDP**	47.85	Light warning
Production Rate of Industry Whole Member Labor	9.51	Middle Warning	**Comprehensive Energy Consume of Per 10000 Yuan GDP**	1.11	Heavy Warning
Percentage of Tertiary Industry's incensement in GDP	48.39	Light warning	**Rate of Total Amount of Foreign Capital Actually Used to GDP**	8.95	Middle Warning
Unemployment Rate	4.83	Heavy Warning	**Engel Coefficient**	41.35	Heavy Warning
Population Natural Growth Rate	1.98	Light Warning	**Population Average Prospective Life Span**	74.36	Middle Warning
Proportion of Non-agriculture Population in City Population	63.65	Light Warning	**Per Capita Floor Space of Houses**	22.33	Light Warning
Percentage of Scientific and Technological、Educational Expenditure to GDP	1.66	Heavy Warning	**Number of Doctors Per 10000 Population**	28.32	Heavy Warning
Number of Books Stored in Public Library Per 100 Person	332.62	Middle Warning	**Green Coverage Rate of City**	43.33	Middle Warning
Per Capita Public Green Area Urban Households	9.34	Middle Warning	**Per Year Density of TSP**	0.14	Middle Warning
Rate of Drinking Water Resources of Reaching Water's Quality Standard	98.78	Light Warning	**Rate of Processed Industrial Waste Water of Reaching Standard**	92.92	Non-warning
Average Value of Regional Environmental Noise	52.03	Light Warning	**Dirty Processing Rate of Water of City**	74.19	Middle Warning
Treatment Rate of Industrial Solid Wastes	87.94	Middle Warning	**Treatment Rate of Industrial Waste Gas**	55.15	Middle Warning
Warning Situation of Nanjing Ecosystem		Light Warning	**Economic sub-system**		Middle Warning
Social sub-system		Light Warning	**natural sub-system**		Light Warning

3. CASE STUDY OF URBAN ECOSYSTEM EARLY-WARNING

Nanjing is a historical cultural city and lies in the Yangtze Rive valley in china. In recent years, its economy grows fast, city's construct is flourishing. However, it suffers some new problems, such as environmental deterioration, resource exhaust, population pressure and so on. In this paper, Nanjing ecosystem as a case is studied, and which can draw lessons to the analogous urban ecosystem.

3.1 Establishing Index System of Early-Warning

The establishment of index system should follow the principles such as systemic combining with hierarchy, comprehensive combining with recapitulative, measure and feasibility, static combining with dynamic and so on. According to the actual of Nanjing Urban ecosystem, 26 indexes of early-warning are chosen, including of the indexes of the structure, level and efficiency in economic system, standard of living and level of education and technology in social system, quality of environment and level of managing environment in natural system. The index system of early-warning are showed in table 1. The data from 1986 to 2004 can obtained from annual.

Both AHP and information entropy are used to acquire the weight of indexes, which include expert's experience and the objectivity information of data. Weight is calculated by AHP and information entropy separately and then the integrated weight is obtained by formula (2) as the weight of indexes [10].

$$w_j = \beta w_{1j} + (1-\beta) w_{2j}, (j = 1,2,\cdots,N) \qquad (2)$$

Where $0 \le \beta \le 1$, generally, β is set 0.5, the weights of indexes are showed in table 1.

3.2 Compartmentalizing Bounds of Early-Warning

In this paper, bounds of Nanjing ecosystem early-warning's index system are compartmentalized by the degrees of sustainable development of its ecosystem, which include non-warning, light warning, middle warning, heavy warning and huge warning. Standard values of bounds are established by criterion and literature [11], which is showed in table 1.

3.3 Results and Analysis

According to the data from 1986 to 2004, the data of each index in 2005 can be forecasted by the gray forecasting method, the results of which are showed in table 2.

When the index is "Huge warning", it gains 1 score. When is "Heavy warning", it gains 2 score. When is "Middle warning", it gains 3 score. When is "Light warning", it gains 4 score. When is "none warning", it gains 5 score. The sun of each index multiplying its weight calculates the integrated index. The bounds of integrated index can be calculated by the method of *ARCH* [12], if the score of integrated index is less than 0.8, the warning condition is "huge warning". If the score is between 0.8 and 2.3, the warning condition is "Heavy warning". If the score is between 2.3 and 3.7, the warning condition is "Middle warning". If the score is between 3.7 and 4.7, the warning condition is "Light warning". If the score is more than 4.7, the warning condition is "None warning". We can obtain the index through computing forecasting data; the results of all early-warning indexes are showed in table 2. The warning degree of economy subsystem, society subsystem and nature subsystem are Middle warning, Light warning, Light warning respectively. Then the warning condition of Nanjing ecosystem can be calculated, which is Light warning.

4. CONCLUSIONS

In this paper, the sustainable development of urban ecosystem is studied. The immune mechanism of urban ecosystem is analyzed and urban ecosystem early-warning model based on artificial immunity is established. Nanjing urban ecosystem as a case and is studied for its sustainable development. The result shows that the model is very effective and it's helpful for the sustainable development of urban ecosystem. Of course, the research of urban ecosystem early warning is complicated, and further studies are certainly desirable.

5. ACKNOWLEDGEMENTS

This work is supported by the Excellent Young Teachers Program (grant no.200384) of Ministry of Education and 948 Program of Ministry of Water Resources (grant no. CT200401) of the People's Republic of China.

6. REFERENCE

[1] F. Vester, V. Hesler, A Ecology and planning in metropolitan areas sensitivity model, Berlin, Federal Environment Agency, 1980.
[2] Odun H T, Systems ecology: an introduction, N Y: John Wiley and Sons, 1982.
[3] S.J. Ma, R.S. Wang, "society-economy–nature compound ecosystem", Journal of Ecology, January 1984, 4(1).
[4] C.L. Gu, "On Research Direction Chinese Urban Continued Development", Urban Planning Forum, June 1994, 94(6): 1-10.
[5] C.C. Xu, M.M. Hou, "Quantitative Evaluation of Urban Sustainable Development", Journal of Resources Development and Market, March 2004, and No.1: 6-8.
[6] X.J. Pu, Z.Z. Xu, "Research on Environment Capacity Indexes and Model Established of Urban Sustainable Development", Journal of Wuhan University, December 2001, 34(6): 12-16.
[7] H.W. Mo, The Principle and Application of Artificial Immune System, Harbin: Harbin industrial university Press, 2002.
[8] D. Dasgupta, S. Forrest, "Artifical Immune Systems in Industrial Applications", Proc. of the IPMM'99, 1999.
[9] P.Z. Li, J.G. Yang, "State Monitoring and Malfunction Diagnosis Based on Self-Noself Distinguish Mechanism", Journal of Shanghai Project and Technology University, September 2004, No.3: 24-27.
[10] J.F. Chen, "Satisfying Solution Model and its Application in Redeploying Water Project Line Scheme Optimal Selection", Journal of Hohai University, January 2004, 32(1): 46-48.
[11] W.Q. Dong, T.M. Gao, The Method of Analyzing and Forecasting the Fluctuation of Economy Periods, Jilin: Jilin University Press, 1998.
[12] H.M. Wang, "Research and Forecasting of ARCH Early-warning System", Forecast, April 1998, No.4: 55-56.

Using Adaptive Genetic Algorithm to Identify Hysteresis

Li Peng, Kaifeng Lu
Control Science and Engineering Research Center, Southern Yangtze University
Wuxi, Jiangsu, 214122 P.R. China
Email: penglimail2002@yahoo.com.cn
And
Wen Li
School of Mechanical Engineering, Southern Yangtze University
Wuxi, Jiangsu, 214122 P.R. China
Email: liwp85@yahoo.com.cn

ABSTRACT

A Genetic Algorithm with Adaptive Search Space (GAASS) is proposed and applied to identify the hysteresis model parameters of an electromechanical-valve actuator installed on a pneumatic system. According to the normalized fitness distance in each generation, the proposed GAASS method consistently identifies the best search domains in the parameter space and adjusts the crossover and mutation rates in order to achieve fast convergence and high accuracy. Experiments have been conducted to investigate the effectiveness of the proposed hysteresis identification approach. The experimental results with three different types of sensors have demonstrated the effectiveness of this proposed method.

Keywords: Adaptive search space, Genetic algorithms, Hysteresis identification.

1. INTRODUCTION

Actuator hysteresis often leads to problems in control systems because it causes tracking errors, limit cycles, and undesired stick-slip motions. Hysteresis exists in electrical valve actuators, mainly due to the ferromagnetic effect associated with the motor drive [10,12]. To counteract hysteresis, a hysteresis model is required in designing a control compensator, and the identification of hysteresis model parameters has become a subject of increasing interests. Conventional model-based identification methods, such as the least square approximation of Preisach models and the interacting multiple model (IMM) approach using a Kalman filter to identify static hysteresis models require derivative calculations of the objective function with respect to hysteresis parameters and are difficult to apply to highly nonlinear and high-dimensional systems [7,11]. Recent research on hysteresis identification has involved stochastic algorithms such as genetic algorithms (GAs), and new methods have been developed to improve the robustness in practical applications [1].

In general, search can be performed with either stochastic search or heuristic search. These two types of search mechanisms feature two unique search strategies: exploration and exploitation [4, 8]. Stochastic search performs search space exploration while ignoring the exploitation of the promising regions of the search space (random search being the extreme case). Heuristic search performs exploitation of solution for possible improvement while ignoring the exploration of the search space (gradient-based algorithms being the extreme case). Maintaining a good balance between exploration and exploitation is essential, as recognized by many researchers [3, 4, 5, 6, 13]. Traditional GAs has a good balance between exploration and exploitation of the search space. However, many of them, especially the Simple Genetic Algorithm (SGA), have very little power in exploiting the best solution [9]. It is better that a heuristic search method is hybridized into the traditional GA, aiming to bridge the gap between stochastic search and heuristic search. Grefenstette proposed a genetic search method with approximate function evaluations to enhance the optimal performance [15]. This method introduces an additional degree of exploitation power to traditional GAs, which helps perform better local exploitation around chromosomes and is able to maintain a good balance between exploration and exploitation in the search space. Moreover, Zoumas, et al employed the real number encoding, which requires much less memory compared to the traditional GAs, which utilize the binary-number encoding [16]. Chen put forward Incremental multiple objective genetic algorithms and made GAs more flexible [17].

In this paper, a Genetic Algorithm with Adaptive Search Space (GAASS) is proposed and implemented for the parameter identification of a selected hysteresis model of an electrical valve actuator. With the proposed GAASS, the search space is adaptively updated to achieve high search accuracy. The best search domains in the parameter space are identified according to the normalized fitness distance in each generation, and the crossover and mutation rates are adjusted accordingly in order to achieve fast convergence and high accuracy. Moreover, the low complexity of the GAASS allows more efficient searching operations. The proposed method has been implemented to identify the Krasnosel'skii's hysteron hysteresis model parameters of an electrical valve actuator installed on a pneumatic system. Three types of sensors (actuator position, air pressure and mass airflow rate) are used to investigate the hysteresis model parameter identification using the proposed GAASS method. The experimental results have demonstrated the effectiveness of the proposed method.

The rest of the paper is organized as follows: Section 2 describes the selected hysteresis model; Section 3 presents the parameter identification procedure using the proposed GAASS method; Section 4 describes the experiment setup and procedures, and presents the experimental results along with discussions; and Section 5 concludes the paper.

2. HYSTERESIS MODEL

Hysteresis model

Hysteresis models have been developed in various ways such as the Bouc-Wen model, Chua-Stromsmoe model,

Preisach model, and Krasnosel'skii's hysteron, etc., as described in [2]. Among them, Krasnosel'skii's hysteron provides a general model of hysteresis, which captures most For our work and the sets of equations to describe Krasnosel'skii's hysteron are derived in Tao and Kokotović [12]:

$$H(v(j)) = u(j)$$

$$= \begin{cases} u(j-1) & \text{if } v(j) = v(j-1) \\ m_{h,t}v(j) + c_{h,t} & \text{if } v(j) \ge v_1, \text{ or} \\ & \text{if } m_{h,t} < m_{h,b}, \\ & u(j-1) = m_{h,t}v(j-1) + c_{h,t} \text{ and} \\ & v(j-1) < v(j) < v_1 \\ m_{h,b}v(j) + c_{h,b} & \text{if } v(j) \le v_2, \text{ or} \\ & \text{if } m_{h,t} > m_{h,b}, \\ & u(j-1) = m_{h,b}v(j-1) + c_{h,b} \text{ and} \\ & v_2 < v(j) < v(j-1) \\ m_{h,t}v(j) + c_d & \text{if } v_d < v(j) < v(j-1) \\ m_{h,b}v(j) + c_u & \text{if } v(j-1) < v(j) < v_u \\ m_{h,l}(v(j) - c_{h,l}) & \text{if } v_d \ge v(j) \ge v_2 \\ m_{h,r}(v(j) - c_{h,r}) & \text{if } v_u \le v(j) \le v_1 \end{cases}$$

$$\text{subject to}: m_{h,l} > 0, m_{h,r} > 0, \min(m_{h,l}, m_{h,r}) > m_{h,t} > 0,$$
$$\min(m_{h,l}, m_{h,r}) > m_{h,b} > 0, c_{h,l} < c_{h,r}, c_{h,t} > c_{h,b} \quad (1)$$

where the eight parameters consist of four slopes mh,l, mh,r, mh,t, mh,b and four crossings ch,l, ch,r, ch,t, ch,b at the left, right, top, and bottom sides of the hysteresis loop as indicated by the subscripts l, r, t, and b, respectively. The index j denotes the sampled data number. The functions v (j) and u (j) denote input and output, and the variables v1, v2, vu, vd are given by:

$$v_1 = \frac{c_{h,t} + m_{h,r}c_{h,r}}{m_{h,r} - m_{h,t}}, \quad v_2 = \frac{c_{h,b} + m_{h,l}c_{h,l}}{m_{h,l} - m_{h,b}} \quad (2)$$

$$v_u = \frac{m_{h,r}c_{h,r} + c_u}{m_{h,r} - m_{h,b}}, \quad v_d = \frac{m_{h,l}c_{h,l} + c_d}{m_{h,l} - m_{h,t}} \quad (3)$$

where in (3), $c_u = u(j-1) - m_{h,b}v(j-1)$ and $c_d = u(j-1) - m_{h,t}v(j-1)$ such that $v_4 \le v_d \le v_u \le v_3$. See the Fig. 1.

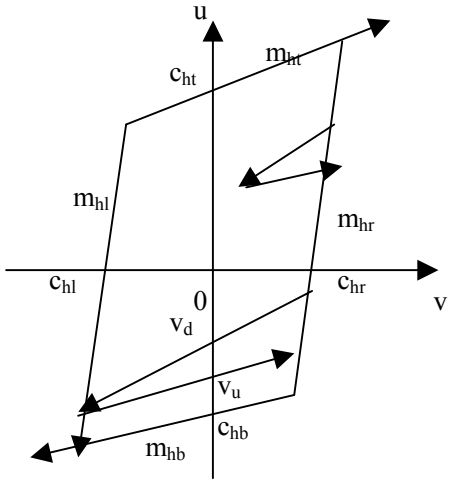


Fig. 1. Hysteretic model

of the hysteretic characteristics and is applicable for parametric inverse compensation. A piecewise linear hysteron model described using eight parameters is selected

Hysteresis Model Parameter Identification Using GAASS

The parameter identification problem is generally defined as identifying a parameter set $x \in \boldsymbol{X}$ when the output data $y \in \boldsymbol{Y}$ and a direct mapping $\boldsymbol{\psi}$: $\boldsymbol{X} \to \boldsymbol{Y}$ is known. The parameter space is defined by:

$$\boldsymbol{X} = \{x \in \boldsymbol{X_1} \times \ldots \times \boldsymbol{X_m} \mid g_i(x) \ge 0, \forall\, i \in \{1, \ldots, m\}\} \quad (4)$$

where m represents the number of parameters and g_i: $\boldsymbol{X_1} \times \ldots \times \boldsymbol{X_m} \to \Re^m$ represents the inequality constraints. Mapping $\boldsymbol{X}$ to $\boldsymbol{Y}$, the input-output relation can be expressed as:

$$y = \boldsymbol{\psi}(x) + e \quad (5)$$

where e represents the error term.

The objective of the proposed parameter identification method is to minimize the following objective function through continuous adaptation of the parameters:

$$J = \sum_{i=1}^{n} |e_i|^2 = \sum_{i=1}^{n} |y_i - \boldsymbol{\psi}_i(x)|^2 \quad (6)$$

Where n is the total number of data points. The adaptation of the system parameters is performed using the proposed GAASS, which searches for a set of parameters that leads to a smaller J until the relative percentage error, e_r%, falls below a predefined error threshold or a prescribed maximum number of generations is reached. The e_r % is given by:

$$e_r\% = \sqrt{\frac{J}{\sum_{i=1}^{n} |y_i|^2}} \times 100\% \quad (7)$$

3. THE GAASS METHOD

The proposed GAASS incorporates an adaptive mechanism in defining the search space in the genetic algorithm (GA). The computation framework of GAASS consists of five major operators: initialization, evaluation, selection, crossover, and mutation. Except for initialization, which only performs once in the entire procedure, all the other four operators execute in every generation until a stop criterion is met. The following describes the various operators of the GAASS including the initialization method, adaptive search space method, selection scheme, and the adaptive crossover and mutation schemes:

Initialization

The initial real-number chromosomes are randomly generated from within the feasible region (constraints) of each system parameter. It uses the rejecting strategy [8] in handling the constraints of the parameters. Rejecting strategy discards all infeasible chromosomes (solutions that violate the constraints) created throughout the evolutionary process. The initial population size, Nipop, is set to twice the size of the population in the latter generations, Npop. Through the tournament selection in the selection operator, only those chromosomes with higher fitness from the initial population are taken to the second generation while those with lower fitness are discarded. This initialization method gives the algorithm a nice start by providing a fine initial sampling of the parameter space.

Evaluation

The evaluation of the fitness function, selection probability pk, and expected value ek of each chromosome, as well as the normalized fitness distance (NFD) [8], is performed in this operator. Fitness function is the same as the objective function J as defined in (6). The selection probability pk and the expected value ek are used for the selection of chromosomes in the selection operator. For chromosome k ∈ [1, Npop] with fitness fk, the values of pk and ek are determined by:

$$p_k = f_k \Big/ \sum_{i=1}^{N_{pop}} f_i \tag{8}$$

$$e_k = N_{pop} \times p_k \tag{9}$$

The normalized fitness distance (NFD) is a measure of the solution convergence. It is analogous to the ratio of the improvement of average fitness to the improvement of the best fitness in a population. It controls the adaptive search space mechanism as well as the exploration and exploitation pressures of the GAASS. NFD is defined by:

$$NFD = \begin{cases} \dfrac{f_{max} - \bar{f}}{f_{max} - f_{min} + \varepsilon} & \textit{for minimization problem} \\ \dfrac{\bar{f} - f_{min}}{f_{max} - f_{min} + \varepsilon} & \textit{for maximization problem} \end{cases} \tag{10}$$

where f_{max} is the maximum fitness value of the population, f_{min} is the minimum fitness value of the population, $\bar{f}$ is the average fitness value, and ε is a small positive number to prevent the equation from zero division. For minimization problem, values of $\bar{f}$ and f_{max} will approach the value of f_{min} as generation progresses. For maximization problem, values of $\bar{f}$ and f_{min} will approach the value of fmax as generation progresses. Thus, the value of NFD will gradually decrease as the solution approaches the optimum value.

If NFD of the present population is smaller than or equal to that of the previous population (solution converges), the size of the adaptive search space, Ω, will be reduced. Otherwise (solution diverges), it will remain the same as that of the previous population. The average of each parameter in the current population, $\bar{x}$, is used as the origin of Ω. The search space Ω ∈ ℜm is then expressed as:

$$\begin{aligned} \boldsymbol{\Omega} &= \prod_{i=1}^{m} \left[\max\{C_{lower,i}, D_{lower,i}\}, \min\{C_{upper,i}, D_{upper,i}\} \right] \\ &= \prod_{i=1}^{m} \left[\max\{C_{lower,i}, (\bar{x}_i - \gamma_i)\}, \min\{C_{upper,i}, (\bar{x}_i + \gamma_i)\} \right] \end{aligned} \tag{11}$$

where D_{lower} is the lower bound of Ω, D_{upper} is the upper bound of Ω, γ is the half search domain, and m is the number of parameters. γ starts with half of the parameter constraint, γi,init, then it gradually reduces dynamically according to the exponential function defined by:

$$\gamma_i(z) = \gamma_{i,init} \exp(-cs) \quad \forall i \in \{1,...,m\}, z \in \{1,...,gen\} \tag{12}$$

where z is the generation count, gen is the maximum number of generation, c is a positive, problem-dependent coefficient, which controls the contracting rate of the search space, and s is the search space index given by:

$$s = \begin{cases} 1 & \text{if } z = 1 \\ s+1 & \text{if } NFD(z) \le NFD(z-1) \quad \forall z \in \{1,...,gen\} \\ s & \text{if } NFD(z) > NFD(z-1) \quad \forall z \in \{1,...,gen\} \end{cases} \tag{13}$$

If D_{lower} falls outside the parameter constraint, it will be replaced by the lower bound constraint C_{lower}. This is also true for D_{upper}, where it will be replaced by the upper bound constraint C_{upper}. This adaptive search space method provides the appropriate search region for the regeneration of chromosomes in crossover and the regeneration of genes in mutation to take place.

Selection

Selection is done using the remainder stochastic sampling. This mixed sampling approach contains both stochastic and deterministic features simultaneously [8]. To further improve the convergence performance, the elitist selection scheme is also used to ensure that the best chromosome is always passed onto the next generation.

Crossover

Crossover is the main search operator in GAs, which performs the exchange of information among chromosomes through combination and disruption of schemata. Investigation in [4] suggests that the essence of effective crossover is to increase both the combination power and the disruption power. In GAASS, the increase of combination power and disruption power is achieved by using an adaptive crossover rate scheme. The number of mating in a population is controlled by the adaptive crossover rate, pc, which changes according to NFD. If NFD of the present population is smaller than or equal to that of the previous population, pc is set to a higher value, pc,h. Otherwise, pc is set to a lower value, pc,l. The adaptive crossover rate scheme is described as:

$$p_c = \begin{cases} p_{c,h} & \text{if } NFD(z) \le NFD(z-1) \quad \forall z \in \{1,...,gen\} \\ p_{c,l} & \text{if } NFD(z) > NFD(z-1) \quad \forall z \in \{1,...,gen\} \end{cases} \tag{14}$$

where the values of pc,l and pc,h are set to 0.5 and 0.9, respectively. According to (14), the number of offspring that will be generated in a population, Nc, is given by:

$$N_c = N_{pop} \times p_c \tag{15}$$

The maximum number of offspring that can be generated in a population is equal to its population size N_{pop}. If the number of offspring generated is fewer than the maximum, the GAASS will perform the chromosome regeneration around the origin of Ω to fill the vacancy according to the following equation:

$$x_{k,i}(z) = \bar{x}_i(z) + r(\gamma_i(z)) \quad \forall i \in \{1,...,m\}, z \in \{1,...,gen\}, k \in \{(N_c+1),...,N_{pop}\} \tag{16}$$

where r ∈ [-1, 1] is a random floating-point number. The adaptive crossover rate scheme ensures that there are

always enough combination power and disruption power in a population to perform effective crossover. When the solution converges, the combination power is increased by the higher crossover rate such that the majority of offspring are produced from the weighted averages of the parents. When the solution diverges, the disruption power is increased by the higher regeneration rate of chromosomes inside Ω such that some new, unbiased chromosomes that could not be produced by the previous genetic operators can be introduced. The role of chromosome regeneration here serves two purposes: First, during early generations where the size of Ω is relatively large and the Euclidean distance between the global optimum and the origin of Ω is relatively long, regeneration performs a more stochastic search in Ω, which hopefully provides better chromosomes for the latter generations. Second, during latter generations where the size of Ω is relatively small and the Euclidean distance between the global optimum and the origin of Ω is relatively short, regeneration performs a more heuristic search in Ω, which helps exploiting the better solution. The crossover procedure of the GAASS employs the two-parent crossover scheme, which is performed by the combined method of the arithmetic crossover and the schema processing similarly as in [1].

Mutation

Mutation serves as a background operator to restore genetic materials as well as a local optimizer since it is a guided-search operator controlled by Ω. Mutation rate is significant in the controlling of the GA performance because it induces diversity to a population and also exploits the better solution. In GAASS, the mutation rate, pm, defined as the number of parameters chosen to mutate in a population is changed adaptively according to NFD. If NFD of the present population is smaller than or equal to that of the previous population (solution converges), pm is set to a higher value, $p_{m,h}$, such that the diversity of chromosomes is increased so to avoid premature convergence. Otherwise (solution diverges), pm is set to a lower value, $p_{m,l}$, since the population already has enough diversity. The adaptive mutation rate scheme is described as:

$$p_m = \begin{cases} p_{m,h} & \text{if } NFD(z) \le NFD(z-1) \quad \forall z \in \{1,...,gen\} \\ p_{m,l} & \text{if } NFD(z) > NFD(z-1) \quad \forall z \in \{1,...,gen\} \end{cases} \tag{17}$$

where the values of $p_{m,l}$ and $p_{m,h}$ are set to 0.01 and 0.3, respectively. According to (17), the number of mutations performed in a population, Nm, is given by:

$$N_m = N_{pop} \times m \times p_m \tag{18}$$

The mutation procedure in the GAASS is performed by randomly generating a value that falls within the subspace of Ω, ω, for each chosen parameter (gene) until the number of mutations has reached the maximum, N_m. The subspace ω defines the range of local optimization and is given by:

$$\boldsymbol{\omega} = \xi \boldsymbol{\Omega} \tag{19}$$

Where ξ is a problem-dependent coefficient, which defines the range of perturbation of each gene. Depending on the parameter sensitivity, the value of ξ usually ranges from 0.01 to 0.05.

The GAASS uses the five operators discussed above to perform an effective search/optimization that contains both stochastic and heuristic characteristics. The mechanism of the adaptive search space method is performed together with the adaptive crossover and mutation techniques to control the pressures of search space exploration and search space exploitation.

The effectiveness of the GAASS method has been validated from the optimization results of three highly-multimodal test functions F1, F2, and F3, which are given respectively as follows [3]:

$$F1(x_1,x_2) = \sum_{i=1}^{5} i\cos[(i+1)x_1 + i] \sum_{i=1}^{5} i\cos[(i+1)x_2 + i];$$
$$x_i \in [-10,10] \forall i \in \{1,2\}; \quad \text{Global optima=186.7307} \tag{20}$$

$$F2(x_1,x_2,x_3,x_4) = \sum_{i=1}^{5} \frac{1}{\sum [(x_j - d(i,j))^2] + c(i)};$$

$$\text{where} \quad d[5,4] = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \end{bmatrix};$$
$$c[5] = [0.1 \quad 0.2 \quad 0.2 \quad 0.4 \quad 0.4];$$
$$x_i \in [0,1.0] \forall i \in \{1,2,3,4\}; \quad \text{Global optimum} = 10.1532 \tag{21}$$

$$F3(x_1,...,x_{20}) = \sum_{i=1}^{20} \left[x_i^2 + 5(1 - \cos(\pi x_i))\right];$$
$$x_i \in [-3,3] \forall i \in \{1,...,20\}; \quad \text{Global optima} = 0 \tag{22}$$

Each test function has been optimized separately using the traditional Simple Genetic Algorithm (SGA) and the proposed GAASS method, and both methods have been run ten times using random seeds from 0 to 9. The parameter settings of the SGA and the GAASS are listed in Table 1. ,

Table 1. Parameter settings of SGA and GAASS

	SGA			GAASS		
Nipop	12	12	40	12	12	40
Npop	6	6	20	6	6	20
Max #generations (*gen*)	100	400	1000	100	400	1000
Max #evaluations	606	2406	20020	606	2406	20020
pc,l	0.5	0.5	0.5	0.5	0.5	0.5
pc,h	0.9	0.9	0.9	0.9	0.9	0.9
pm,l	0.05	0.05	0.01	0.05	0.05	0.01
pm,h	0.5	0.5	0.1	0.5	0.5	0.1
Ω contraction rate (*c*)	N/A	N/A	N/A	0.05	0.01	0.005
Subspace coefficient (*ξ*)	N/A	N/A	N/A	0.02	0.05	0.05
Function Solving	*F*1	*F*2	*F*3	*F*1	*F*2	*F*3

and the convergence performance comparisons are shown in Figs. 2 ~ 4. Note that only the best of the ten SGA optimization results is shown for each function. As shown from the convergence comparisons in Fig. 2.~ 4., the GAASS method has consistently outperformed the

SGA in both the convergence speed and the solution accuracy.

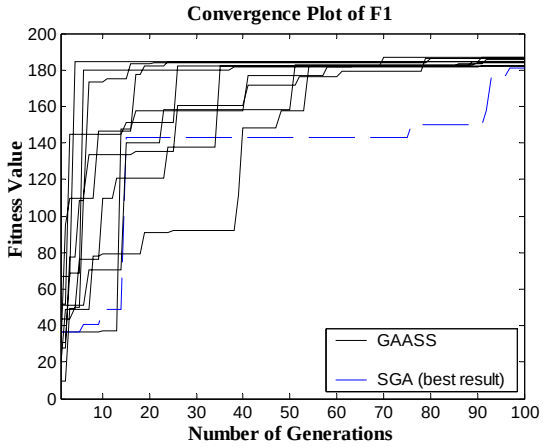


Fig. 2. *F*1 convergence comparison

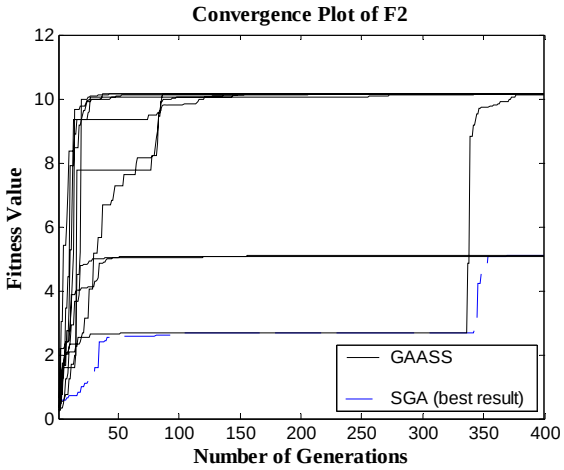


Fig. 3. F2 convergence comparison

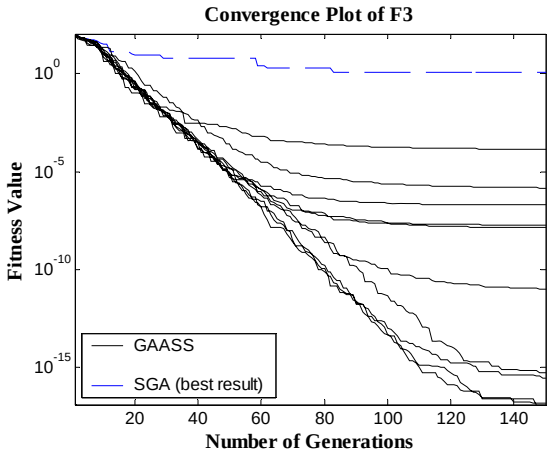


Fig. 4. *F*3 convergence comparison

4. CONCLUSIONS

A new adaptive search technique is incorporated with a genetic algorithm in the proposed GAASS method, which is then successfully applied to actuator hysteresis model parameter identification and inverse compensation. The proposed GAASS approach has efficiently identified the hysteresis model parameters induced to an electrical valve actuator. The hysteresis inverse compensation based on the model using the identified hysteresis parameters has effectively softened the effect of actuator hysteresis.

5. REFERENCES

[1] C. H. Chan, G. Liu, "An adaptive search space based evolutionary algorithm with application to actuator hysteresis identification", Proceedings of the 2003 American Control Conference (ACC-03), Denver, CO, 2003, 2175-2180.

[2] L.O. Chua, S.C. Bass, "A generalized hysteresis model", IEEE Trans. Circuit Theory, CT-19, 1972, 36 -48.

[3] K.A. De Jong, W.M. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms", Annals of Mathematics and Artificial Intelligence, 1992, 1-26.

[4] A.E. Eiben, C.A. Schippers, "On evolutionary exploration and exploitation", Fundamenta Informaticae, 1998, 35(1-4): 35-50.

[5] L.J. Eshelman, R.A. Caruana, et al, "Biases in the crossover landscape", Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann, 1989.

[6] L.J. Eshelman, J.D. Schaffer et al "Crossover's niche", Proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann, 1993, 9-14.

[7] W.S. Galinaitis, D.S. Joseph et al, "Parameter identification for Preisach models of hysteresis", Proceedings of DETC 2001 ASME Design Engineering Technical Conferences, Pittsburgh, PA, 2001, 1-9.

[8] M. Gen, R. Cheng, "Genetic algorithms and engineering design", John Wiley & Sons, Inc., New York, NY, 1996.

[9] D. E. Goldberg, "Genetic algorithms in search", optimization and machine learning, Addison-Wesley, Reading, MA, 1989.

[10] M.A. Krasnoselskii, A.V. Pokrovskii, "Systems with hysteresis", Springer-Verlag, Berlin, Germany, 1983.

[11] L. Mihaylova, V. Lampaert, et al "Identification of hysteresis functions using a multiple model approach", International Conference on Multisensor Fusion and Integration for Intelligent Systems, Baden-Baden, Germany, 2001.

[12] G. Tao, P.V. Kokotović, "Adaptive control of systems with actuator and sensor nonlinearities", John Wiley & Sons, Inc., New York, NY, 1996.

[13] S. Tsutsui, A. Ghosh, et al, "A real-coded genetic algorithm with an explorer and an exploiter populations", Proceedings of the 7th International Conference on Genetic Algorithms, Morgan Kaufmann, 1997, 238-245.

[14] Y.G. Xu, G.R. Liu et al, "A Novel Hybrid Genetic Algorithm Using Local Optimizer Based on Heuristic Pattern Move", Applied Artificial Intelligence, 2001, (15): 601-631.

[15] J. J.Grefenstette, J.M.Fitzpatrick, "Genetic search with approximate function evaluations", Proceedings of an International Conference on Genetic Algorithms and Their Application, 1985, 112-120.

[16] C.E.Zoumas, A.G.Bakirtzis, et al, V.,"A genetic algorithm solution approach to the hydrothermal coordination problem", IEEE Transactions on Power Systems, 2004, 19(3): 1356-1364.

[17] Q. Chen, S. U. Guan, "Incremental multiple objective genetic algorithms", IEEE Transactions on Man and Cybernetics Systems, Part B, 2004, 34(3): 1325-1334.

A Novel Algorithm for Vehicle License Localization Based on Wavelet Transform

Yonghui Pan 1,2
[1]School of Information Technology, Southern Yangtze University
[1]No. 1800, Lihudadao, Wuxi Jiangsu 214122, China
[2]Jiangyin polytechnic college
Email: pyh@mail.jypc.org
And
Fang Bao1,2
1School of Information Technology, Southern Yangtze University
No. 1800, Lihudadao, Wuxi Jiangsu 214122, China
2Jiangyin polytechnic college
And
ShiTong Wang1
1School of Information Technology, Southern Yangtze University
No. 1800, Lihudadao, Wuxi Jiangsu 214122, China

ABSTRACT

The paper presents a novel algorithm is proposed for detecting vehicle license, based on wavelet transform. Firstly, it adopts the mean, energy and entropy on the wavelet coefficients of color vehicle image as textural features. Secondly, an algorithm based on dynamic threshold is chosen to characterize license and non-license areas. Thirdly, a morphological operation was used to classify the image into license candidates and background. Finally, the detected text candidates undergo the empirical-rules analysis to identify license area. Experimental results demonstrate that the proposed approach could efficiently be used as a vehicle license detection system, which is robust for license-size, license-color and background complexity.

Keywords: License Localization, Wavelet Transform, Wavelet Coefficients, C-Means.

1. INTRODUCTION

Since vehicle images are usually photographed in complex background influenced by various illumination, it's rather difficult to detect license plate. Therefore, it's becoming a key issue as to how to accurately locate vehicle license region. With the anticipation to resolve the above-mentioned problem, people have, based on tremendous researches, proposed numerous localization algorithms, most of which are based on the different features of the vehicle license.

Among these algorithms, the relatively matured are listed as follows: Edge detection algorithms use the inverted form of "L", horizontal and vertical aligned strokes to detect and locate characters of license, thus finding out a group of characters complying with certain rules, which is supposed to be a license plate [1]. Region growing algorithms are applied to grow the edge of image evenly so as to capture license candidates, then by using the geometry features of license and gray histgram of license edge, the fake license candidates are discarded while the real one is located[5,9]. Morphological algorithms for gray images employ license features in shape and in character arrangement to undergo morphological operations for the preprocessed gray image so as to figure out the license region in which straight lines are corresponding to a certain number of characters [4, 10, 13, 14]. Segmentation algorithms based on string-feature intensification use linear filter to intense the texture of license, and then segment the license by threshold [9, 10, 13]. Fuzzy clustering algorithms use fuzzy logic system to detect license from license candidates by some cluster parameters [3,5,11,15]. License localization and segmentation for a gray image uses one iteration method to capture its binary image through a proper threshold, and then locate its license according to vertical edge features of the strokes of characters [2].

Although the algorithms of license localization above have taken some effects in practical appliances, the accuracy is still not high enough for real time supervising vehicle. Therefore, a novel algorithm of license localization based on wavelet transform is proposed in this paper, which can greatly improve the accuracy of license localization. The principle is: Firstly, the color vehicle image is transformed into index image which then undergoes wavelet transform. So we can further analyze the wavelet coefficients of LH sub-band. Secondly, wavelet coefficients can be classified into two groups through dynamic threshold, which was obtained through the mean, energy and entropy of the coefficients of LH sub-band. By applying morphological operations, these two clusters form two areas: license candidates and non-license candidates. At last, the detected license candidates undergo the empirical rules to identify license areas and project profile to refine their location.

The paper is organized as follows. Section 2 presents the individual steps of our approach for license localization. Section 3 discusses the experiment results. In the final section conclusions are given.

2. LICENSE DETECTION

In this section, the processing steps of the proposed approach are presented. Our aim is to build an automatic license detection system, which is capable of handling automobile image with complex background, little slope, arbitrary size and color. From Fig. 1. we can see that the proposed approach is mainly performed by four steps: wavelet transform, texture feature extraction, license candidates detection, license localization, which will be

described in detail as follows.

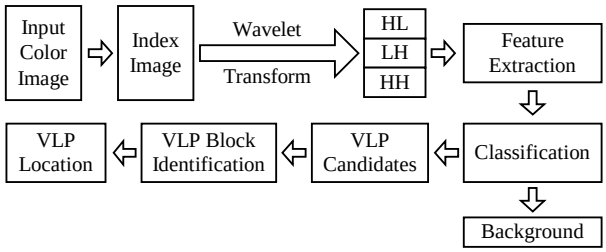


Fig. 1. Flow chart of the proposed approach

2.1 Wavelet Transform of the Image

Vehicle license plate (VLP) is mainly composed of about 50 Chinese characters, 26 English capitalization characters and 10 Arabian figures, and the font of the VLP is stable. A VLP consists about 7 characters and forms a rectangle whose ratio of width and height is 4:1. Text is mainly made up of the strokes in horizontal, vertical, up-right, up-left direction. It has weak and irregular texture property, and can be done as a special texture [18]. Texts in the same VLP often have the same color, and contrast clearly with the background, which also have the same color. There are “blue background white character”, “yellow background black character”, “black background white character” and “white background black character” ect.[17]. So the edge information of the VLP is abundant [17, 18]. These texture properties and color features are combined to locate VLP in vehicle images. Firstly, we convert the color vehicle image into indexed image. Then on the indexed image a wavelet transform is applied to capture coefficients of respective sub-bands.

The main characteristic of wavelet transformation is to decompose a signal into sub-bands at various scales and frequencies, which is useful to detect edges with different orientations. In the 2-D case, when the wavelet transform is performed by a low filter and a high filter, four sub-bands are obtained after filtering: LL (low frequency), LH (vertical high frequency), HL (horizontal high frequency) and HH (high frequency). In the three high-frequency sub-bands (HL,LH,HH), edges in horizontal, vertical and diagonal directions were detected. Since license area is commonly characterized by strong contrast edges, high valued coefficients can be found in the high-frequency sub-bands. From Fig. 2. we can see it.

（Origin Image）

（HL）

（LH）

（HH）

Fig. 2. Wavelet coefficients of vehicle image

2.2 License Feature Extraction

As we know, the shape of license is a horizontal rectangle. There are about 7 characters and 20-25 pieces of high contrast, dense edge information in a license area. From Fig. 3. we can see it. So high valued coefficients can be found in the vertical high frequency sub-bands (LH) and the coefficients form a dense block.

Fig. 3. Edge information of license

In this paper, license feature extraction is based on LH (vertical high frequency sub-bands) of wavelet coefficients of a vehicle image. Since the value of coefficients in license area is very high, we can filter the low valued coefficients through a dynamic threshold. And how to select an effective threshold is the key to filtering successfully. Here, we employ the statistical features in mean, energy and entropy on the transformed vehicle image to capture the dynamic threshold [6,12,18]. They are computed using the formula as follows:

$$f_1 = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} I(i, j) \quad \text{Eq. (1)}$$

$$f_2 = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} I^2(i, j) \quad \text{Eq. (2)}$$

$$f_3 = \sum_{i,j} I(i, j) \cdot \log I(i, j) \quad \text{Eq. (3)}$$

Where, I is the license, w is the width of the image, h is the height, and (i, j) is the pixel position in the image. The dynamic threshold is computed using the formula as follows:

$$K_m = a_m \cdot f_m \quad \text{Eq. (4)}$$

Where K_m is the threshold in various situations, a_m is the respective coefficient of K_m, and f_m is the statistical features of formula (1), (2), (3), and, m =1，2，3. To accelerate the computation speed, the statistical feature of mean and energy is chosen to capture the dynamic threshold. From Fig. 4. we can see the filtered wavelet coefficients (LH, HH).

（LH）

（HH）

Fig. 4. Filtered wavelet coefficients

2.3 License Candidates Detection

When we detect the license in the image, it could be assumed that the vehicle image is composed of two clusters: license area and background. Seen from Fig. 4., high valued coefficients can be found in the vertical high frequency sub-bands (LH) and the coefficients form a dense block. Firstly, a binary process is applied to the remained coefficients of LH sub-band. Then two morphological operations “dilation” and “open” are applied to the transformed image (LH sub-band). The result can be seen from Fig. 5. (a). Finally, a sliding window of size $k \times l$ pixels is moved over the transformed image to discard small isolated objects as background and reserve the corresponding license candidates. The results of the license candidates were shown after morphological operation in Fig. 5. (b).

（a）LH on dilate and fill operation

（b）LH on open operation

Fig. 5. Wavelet coefficients after morphological operation

2.4 License localization

License localization is mainly performed by three steps based on some empirical knowledge:

Step 1, the algorithm of region growing is used to compute the ratio of width, height and areas of all license candidates. If the ratio or areas is too big or small, we think the candidate is background and discard it.

Step 2, according to the stable property of the couples of license characters and background, such as “blue background white character”, “yellow background black character”, “black background white character” and “white background black character”, we apply C-means algorithm to cluster the pixels of all license candidates based on color-couple[7,8,11,15,16,17], and compute the pixel number of every cluster.

Step 3, almost all of the position of the license in a vehicle image is lower than half of the image height. We can compute the height of the license candidate from itself to the bottom of the vehicle image to locate the license.

If any of the above steps locates the license, license detection task has been accomplished.

The rules are noted as follows:

--- the ratio of width and height of license candidates (t)： $t_1 < t < t_2$

--- the areas of license candidates (S)： $S_1 < S < S_2$

--- the pixel number of clusters based on color couple (c)： $c > c_1$

--- the height of license candidates in vehicle image

(h): $h < h_1$

Here, t_1, t_2, s_1, s_2, c_1, h_1 is respective threshold for the above rules. According to experiments, they are appropriate as noted in Table 1.

Table 1. Parameters of empirical rules

t_1	t_2	s_1	s_2	c_1	h_1
2	5	1000	8000	$s/2$	$H/2$

In Table 1., H is the height of vehicle image.
If the license candidate does not satisfy the rules, it is considered to be a non-license area. On the contrary, the candidate is license if it satisfies the rules. Finally, the license undergoes a project profile analysis to refine the license location, and is labeled with the red rectangle to circle the license. From Fig. 6. , the results of some samples are shown.

3. EXPERIMENTAL STUDIES

In order to evaluate the proposed approach described here, a dataset of 200 vehicle images was obtained in various backgrounds. We preprocessed all the images into nearly 200000 pixels (500×400) depending on the original image size to save computation costs. A bior3.7 wavelet was applied to transform the vehicle image. From Table 2, we can see the value scope of threshold coefficients a_m in formula (4). Parameters k =60 and l =20 are introduced to scan the image, since the size of license candidate is more than that of the rectangle of 60×20.

Table 2. Scope of threshold coefficients

a_1	a_2	a_3
5-10	1.5-3	10-18

After processed by step 1 of license localization, the accuracy of license localization is very high. And the accuracy is nearly 100% with the process of step 2 & 3. The reference "A New License Plate Locating Approach Based on Texture and Color Characteristics" used the texture and shape feature of license plate based on region growing to extract license repeatedly, and great effect had been achieved. The accuracy of once localization is 96%. Further more, a color segmentation based on color feature of license is proposed. It made the accuracy of license localization amount to 99.6%. In this paper, the once accuracy of the license localization approach based on wavelet transform is nearly 98%. The further accuracy of license localization amounts to nearly 100%. The experimental results tested by the feature of mean, energy and entropy can be seen in Table 3. It demonstrates, on license localization, the accuracy and robust of our algorithm is far better than that of any other algorithm.

Table 3. Detection results by different feature in various threshold coefficients

Feature	Parameter (a_m)	Step-1 Precision (%)	Step-2&3 Precision (%)
Mean	7	98.8	99.9
	8	99.3	100
	9	98.2	99.9
Energy	1.5	97.6	99.8
	2	98.5	100
	2.5	98.2	99.9
Entropy	15	96.8	98.8
	16	97.7	99.2
	17	95.2	97.8

Fig. 6. Shows the experimental results of various kinds of vehicle images with different color, size, illumination and background. Fig. 6. (a) shows a "blue background white character" license of black car with background of a bungalow, and the size of license is 145×33. Fig. 6. (b) shows a "blue background white character" license of a white car with background of a building site, and the size of license is 149×35. Fig. 6. (c) shows a "blue background white character" license of a white bus with background of a parking center, and the size of license is 94×25. Fig. 6. (d) shows a "yellow background black character" license of a truck with background of road, and the size of license is 105 × 26. The experimental results demonstrate the proposed approach is robust for the different color, size, illumination and background of vehicle license.

(a)

(b)

(c)

（d）

Fig. 6. Experimental results

4. CONCLUSIONS

In this paper, a novel algorithm based on wavelet transform is presented for license localization. In general, the color of character and background in a license are respectively the same and form several stable color-couples. So strong contrast, dense edge information in a license area is abundant. According to the texture characteristics of the license characters and color contrast between texts and background of license in the vehicle images, the color texture features are applied to detect the license in the vehicle images based on wavelet transform. The experimental results with various kinds of the vehicle images demonstrate that the proposed method is effective to locate license automatically in a vehicle image. It is robust for license size, license color, background complexity and various illuminations. The algorithm is executed at a computer of P4-2G, 256M taking nearly 0.56 seconds. So it needs to improve the calculation speed.

To test the effectiveness of the algorithm, we also apply this method in the moving vehicle images to locate the license, and the experimental results are satisfactory. It is found that the proposed algorithm also meets difficulties in few vehicle images with strong variational illumination and badly distortion. In addition, the linear combination and non-linear combination of mean, energy and entropy can be used to capture the threshold, which will optimize the algorithm and improve the accuracy of license localization. For these shortcomings, these problems need to be tackled in future researches.

5. REFERENCES

[1] N.Wan, J.Li, B.Ren et al, “Segmentation Method of Multi-level Automobile License Localization”, Journal of Anhui University Natural Science Edition, 1996, 23(2).

[2] W.Wei, X.H.Huang et al, “A Method of Number-plate Locating Based on The Vehicle Edge Characteristics of Characters”, China Journal of Highway and Transport, 2000, 13(4).

[3] W.Z.Zhou, X.H.Feng et al, “Clasification Based on Feature Extraction from Cluster of Wavelet Coefficients”, Journal of Computer Research & Development, 2001, 38(8).

[4] X.P.Li, J.X.Ren et al, “Study on Automobile License Plate Identification System”,Journal of Beijing Institute of Technology, 2001, 21(1).

[5] K.Q.Cheng, M.J.He, “A review on the chromatic segementation”, Journal of Hubei Normal University (Natural Science), 2004, 24(4).

[6] W.Wu, “Multilevel thresholding algorithm for image segmentation based on maximum fuzzy entropy”, Systems Engineering and Electronics, 2005, 27(2).

[7] H.F.Hou, S.H.Liu, “An Improved Fuzzy C-means Algorithm Based on Genetic Algorithm”, Computer Engineering, 2005, 31(17).

[8] W.J.Li, D.Q.Liang et al, “A Novel Approach for Vehicle License Localization Based on Color-Edge Pair”, Chinese Journal of Computers, 2004, 27(2).

[9] X.H.Fan, F.H.Qi, “A New License Plate Locating Approach Based on Texture and Color Characteristics”, Computer Engineering, 2004, 30(13).

[10] W.J.Li, D.Q.Liang et al, “Segmentation Method for Car Emblem Based on Texture Homogeneity Measure”, Application Research of Computer, 2004, (10).

[11] Q.F.Zhang, Y.W.Cen, et al, “Comprehensive Review of License Plate Localization Technology Based on Color”, Journal of Anhui University of Technology, 2005, 22(1).

[12] F.Yan, Z.P.Lei, “Research of automatic registration technique of the license plate based on the dyadic operation”, Journal of Wuhan Institute of Chemical Technology, 2004, 26(1).

[13] Y.C.Li, H.Y.Sun et al, “An Approach for Vehicle License Plate Locating”, Journal of Institute of Command and Technology, 2001, 12(6).

[14] J.Yang, X.Li et al, “A Recogniton of Vehicle Plate with Mathematical Morphology and Neural Networks”, Journal of Wuhan University of Technology (Transportation Science & Engineering), 2001, 25(3).

[15] X.C.Zhao, F.H.Qi, “Automatic Recognition of Vehicle License Based on Color Segmentation”, Journal of Shanghai Jiaotong University, 1998, 32(10).

[16] X.F.Shi, “Automatic Car Plate Reading System Based on Color Image Processing”, Computer Engineering & Science, 2005, 27(9).

[17] Y.Liu, X.Q.Chen et al, “A Character Segmentation Algorithm of Color Vehicle License Plate Using Color Information”, Computer Applications and Software, 2005, 22(8).

[18] C.M.Liu, C.H.Wang et al, “Text Detection in Images Based on Color Texture Features”, Advances in Intelligent Computing, 2005. Proceedings. 2005 International Conference on, 2005, 1:40-48.

A Multi-Phased Quantum-Behaved Particle Swarm Optimization algorithm

Wenbo Xu, Chunyan Zhang

School of Information Technology, Southern Yangtze University, Wuxi Jiangsu 214122,China

Email:zcy0006@163.com

Abstract:

Quantum-behaved Particle Swarm Optimization (QPSO) is a novel variant of particle Swarm Optimization algorithm. In contrast to Standard Particle Swarm Optimization (SPSO), QPSO guarantees that particles converge to global optimum point in probability and the algorithm has better performance. This paper based on Sun's work [1, 2] introduces an improved Quantum-behaved Particle Swarm Optimization, Multi-Phased QPSO (MQPSO), in which the swarm evolves with multi-swarm and multi-phase. It avoids particle pre-maturity and improves global search performance. The results of several important test functions confirm that the convergent performance of MQPSO outperforms PSO and QPSO in general.

Keywords: PSO, QPSO, Global Convergence, Premature

1 INTRODUCTION

Particle Swarm Optimization (PSO) algorithm is a population-based evolutionary computation technique. The method was originally proposed by J. Kennedy and R.C. Eberhart as a simulation of social behavior of bird flock and was initially introduced as an optimization method in 1995 [3]. PSO can be easily implemented and it is computationally inexpensive, since its memory and CPU speed requirements are low. One reason is that the method requires only the function value and does not require gradient information of the objective function of the Global Optimization problem under consideration. Another reason is that the primitive mathematical operators are used.

However, as demonstrated by F. Van Den Bergh, PSO is not a global convergence guaranteed algorithm because the particle is restricted to a finite sampling space for each of the iterations. This restriction weakens the global search ability of the algorithm and may lead to premature convergence in many cases. To overcome the shortcomings of the PSO, early concept of a Quantum-behaved Particle Swarm Optimization (QPSO) was disseminated in conference reports such as [1], [2].

To improve the global search ability of QPSO, in this paper, we propose a Multi-Phased QPSO algorithm, in which the particle swarm is divided into several sub-swarm and each sub-swarm evolves in the different phase. The MQPSO has been proved to be superior to PSO and QPSO in avoiding premature convergence.

2 QUANTUM-BEHAVED PSO

Particle Swarm Optimization (PSO), a population-based random search technique, originally proposed by J. Kennedy and R. Eberhart [3], has become a most fascinating branch of evolutionary computation. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory. In the Standard PSO with population size *M*, each individual is treated as a volume-less particle in the n-dimensional, with the position and velocity of i^{th} particle represented as $X_i = (X_{i1}, X_{i2}, \cdots, X_{in})$ and $V_i = (V_{i1}, V_{i2}, \cdots, V_{in})$. The particle moves according to the following equation:

$$V_{id} = w \cdot V_{id} + c_1 \cdot r_1 \cdot (P_{id} - X_{id}) + c_2 \cdot r_2 \cdot (P_g - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id}, \quad (d = 1,2,\cdots,n) \quad (2)$$

where c_1 and c_2 are called acceleration coefficients and r_1 , r_2 are two numbers distributed uniformly in [0,1], i.e. $r_1, r_2 \sim U(0,1)$. Parameter w is the inertia weight introduced to accelerate the convergence speed of PSO. Vector $P_i = (P_{i1}, P_{i2}, \cdots, P_{in})$ is the best previous position (the position giving the best fitness value) of particle *i* called personal best position (***pbest***), and vector $P_g = (P_{g1}, P_{g2}, \cdots, P_{gn})$ is the position of the best particle among all the particles in the population and called global best position (***gbest***).

In Quantum-behaved Particle Swarm Optimization (QPSO), the particle moves according to the following equation:

$$p_{id} = \varphi \cdot P_{id} + (1-\varphi) \cdot P_{gd}, \quad \varphi = rand() \quad (3)$$

$$X_{id} = p_{id} \pm \alpha \cdot |mbest_d - X_{id}| \cdot \ln(1/u), \quad u \sim U(0,1) \quad (4)$$

where *mbest* is the mean best position among the particles. That is

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \ \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \ \cdots, \ \frac{1}{M}\sum_{i=1}^{M} P_{in} \right) \quad (5)$$

The p_{id}, a stochastic point between P_{id} and P_{gd}, is the local attractor on the d^{th} dimension of the i^{th} particle, φ is a random number distributed uniformly in [0,1], u is another uniformly-distributed random number in [0,1] and α, called Contraction-Expansion Coefficient, is a parameter of QPSO. The Quantum-behaved Particle Swarm Optimization (QPSO) Algorithm in [1], is described as follows.

1. Initialize an array of particles with random positions inside the problem space;
2. Determine the mean best position among the particles by Eq(5);
3. Evaluate the desired objective function (take minimization problems for example) for each particle and compare with the particle's previous best values: If the current fitness value is less than previous best value, then set the previous best value to the current value. That is, if $f(X_i) < f(P_i)$, then $X_i = P_i$;
4. Determine the current global position minimum among the particle's best positions. That is: $g = \arg\min_{1 \le i \le M}(f(P_i))$ (for maximization problem);
5. Compare the current global position to the previous global: if the fitness value of the current global position is less than that of the previous global position; then set the global position to the current global;
6. For each dimension of the particle, get a stochastic point between P_{id} and P_{gd} by Eq(3);
7. Update position by stochastic Eq(4);

Repeat steps 2-7 until a stop criterion is satisfied **OR** a pre-specified number of iterations are completed.

3 MQPSO

Like PSO, pre-mature in QPSO is also inevitable. In PSO, when the best particle position *pbest* of the particle is very close to the gbest position, the particle may search in a very small region. When the distance between the *pbest* and the *gbest* is close to 0, it can be seen that the speed of the particle will be also close to 0 by equation (1). In QPSO, it means that the parameter L of the particle is very small when the *pbest* is near to the *gbest*. So the searching space of the particle also becomes very small, which may result in search stagnation. If the *gbest* is a local optimum solution in this time, the whole particle swarm will tend to premature convergence because all particles become more and more lack of energy.

To enhance the convergence performance of algorithm, in our proposed QPSO, the particle swarm divided into several sub-warms and each small sub-swarm may search in different phases.

It has been proved that when coefficient β <1.78, particle will converge to *gbest* position, otherwise the particle will diverge [4]. In MQPSO, the particles swarm was divided into two groups, each group may evolves in different phases alternatively. In the first group in phase 1, coefficient is set as $\beta = 0.72$, and the particle flies in attraction mode; When in phase 2, coefficient $\beta = 2$, particle moves in explosion mode. For the second group, when in phase 1, β is set to 1.8, particle flies in explosion mode; in phase 2, setting coefficient $\beta = 0.72$,particle will converge.

The coefficient selection strategy is outlined as follows:

phase 1,group1: $\beta = 0.72$

phase 1,group2: $\beta = 1.8$

phase 2,group1: $\beta = 2$

phase 2,group2: $\beta = 0.72$

Thus the MQPSO can be described as follows.

Initialize the position vector of the particle in the population

Do

For each particle of group1

Use equation to work out mbest1

If phase=1

$\beta = 0.72$

If phase=2

$\beta = 2$

For 1:dimensionD

$\varphi_1 = rand(0,1), \varphi_2 = rand(0,1)$

$p = (\varphi_1 * p_{id} + \varphi_2 * p_{gd})/(\varphi_1 + \varphi_2)$

u=rand(0,1)

if rand(0,1)>0.5

$x_{id} = p - \beta * |mbest - x(t)| * \ln(1/u)$

else

$x_{id} = p + \beta * |mbest - x(t)| * \ln(1/u)$

For each particle of group 2

Use equation to work out mbest2

If phase=1

$\beta = 1.8$

If phase=2

$\beta = 0.72$

For 1:dimensionD

$\varphi_1 = rand(0,1), \varphi_2 = rand(0,1)$

$p = (\varphi_1 * p_{id} + \varphi_2 * p_{gd})/(\varphi_1 + \varphi_2)$

u=rand(0,1)

if rand(0,1)>0.5

$x_{id} = p - \beta * |mbest - x(t)| * \ln(1/u)$

else

$x_{id} = p + \beta * |mbest - x(t)| * \ln(1/u)$

Until the ending condition is met.

4 Results and Analysis

In order to test the performance of MQPSO, four benchmark functions are used, the minimum of these four functions are zero. The QPSO and SPSO are also tested for performance comparison.

1 Sphere Function

$$f(x)_1 = \sum_{i=1}^{n} x_i^2 \tag{6}$$

2 Rosenbrock Function

$$f(x)_2 = \sum_{i=1}^{n}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (7)$$

3 Griewank Function

$$f(x)_3 = \frac{1}{4000}\sum_{i=1}^{n}(x_i - 100)^2 - \prod_{i=1}^{n}\cos(\frac{x_i - 100}{\sqrt{i}}) + 1 \quad (8)$$

4 Shaffer Function

$$f(x)_4 = 0.5 + \frac{(\sin\sqrt{x^2 + y^2})^2 - 0.5}{1.0 + 0.001(x^2 + y^2)^2} \quad (9)$$

Table 1. Initializing area of each testing function

Functions	Initialization Ranges
f_1	(50,100)
f_2	(15,30)
f_3	(300,600)
f_4	(30,100)

Table 2. The upper limit values to each Testing Function

Functions	$X_{\max}$	$V_{\max}$
f_1	100	100
f_2	100	100
f_3	600	600
f_4	100	100

For each Testing Function, we separately use different size of particle swarm and different variable dimension. The size of the first three are 40 and 80, the dimensions are 10,20and 30, respectively. The maximum numbers of iterations are 1000, 1500 and 2000, respectively. The dimension of the last function is 2 and the maximum number of iteration is 2000. The initial ranges of each algorithm are shown in table1. $V_{\max}$ and $X_{\max}$ of each algorithm are shown in table2..The best 50 times average fitness of each function are shown in table3., 4., 5. and 6..

Table 3. The average fitness of the spare function

M	D	$G_{\max}$	SPSO	QPSO	MQPSO
40	10	1000	1e-23	1e-41	8.8581e-073
	20	1500	1e-14	1e-23	3.3765e-053
	30	2000	1e-10	1e-14	5.2859e-043
80	10	1000	1e-28	1e-61	5.5880e-085
	20	1500	1e-17	1e-32	1.8136e-068
	30	2000	1e-12	1e-19	1.9944e-057

Table 4. Shows the average fitness of the rosenbrock function

M	D	$G_{\max}$	SPSO	QPSO	MQPSO
40	10	1000	70.2139	15.8623	57.7131
	20	1500	180.9671	112.4612	84.7484
	30	2000	299.7061	76.4273	150.8813
80	10	1000	36.2954	36.3405	29.6482
	20	1500	87.2802	23.5443	28.2592
	30	2000	205.5596	71.9221	44.7947

Table 5. Shows the average fitness of the griewank function

M	D	$G_{\max}$	SPSO	QPSO	MQPSO
40	10	1000	0.0862	0.0484	0.0022
	20	1500	0.0286	0.0004	0.0011
	30	2000	0.0127	0.0009	4.0765e-005
80	10	1000	0.0760	0.0000	2.6676e-005
	20	1500	0.0288	0.0000	6.8011e-006
	30	2000	0.0128	0.0000	2.0509e-006

Table 6. Shows the average fitness of the shaffer function

M	D	$G_{\max}$	SPSO	QPSO	MQPSO
40	2	2000	0.0006	0.0018	8.5010e-004
80	2	2000	0.0002	0.0004	2.1490e-004

By comparing the experiment results, the global search ability of MQPSO is better than QPSO and SPSO. Therefore, on the base of QPSO, particle swarm is divided into multi-particle swarm and multi-phase particle swarm, the ability of global convergence is greatly improved.

5. Conclusion

In this paper, Quantum-behaved Particle Swarm Optimization is proposed, it guarantees that particles converge in global optimum point in probability, the evolution equations are quite different from SPSO, and parameters are less, just β, which is used to control the converge velocity. However, QPSO has disabilities; the particle swarm tends to pre-maturity convergence. Hence, particle swarm is divided into multi-particle swarm and multi-phase particle swarm, the ability of global convergence is greatly improved.

[1] J. Sun, W.B.Xu, A Global Search Strategy of Quantum-behaved Particle Swarm Optimization[C]. Proceedings of IEEE conference on Cybernetics and Intelligent Systems. (2004):111 – 116

[2] J.Sun, B.Feng, et al, Particle Swarm Optimization with Particles Having Quantum Behavior[C]. Proceedings of 2004 Congress on Evolutionary Computation. (2004): 325-331

[3] J Kennedy, R. C. Eberhart, “Particle Swarm Optimization”. Proceedings of the IEEE International Joint Conference on Neural Networks, 1995,4:1942-1948.

[4] W. B. Xu, J. Sun, “Adaptive Parameter Selection of Quantum-Behaved Particle Swarm Optimization on Global Level,” ICIC 2005, Springer-Verlag Berlin Heidelberg 200

The Research of Short-term Electricity Price Intelligent Forecast System Based on Multi-Agent

Wei Jiang, Zuzhi Shen, Hong Zhang, Fuzhong Wang, Weibin Ding
School of Management, Zhejiang University
Hangzhou, 310058, China
Email: bestof@163.com

ABSTRACT
The short-term electricity price forecast can not be accomplished by only one single forecast method or model and manual proactively participation is utmost important. An intelligent forecasting system built in three basic models, combination model, and expert model is proposed. Through hierarchical analyzing of the "agent" architecture and using the IDEF1 method, the overall architecture of the forecast system based on "multi-agent" was constructed. Based on the composition of "multi-agent" system architecture and "knowledge base", a forecast work process based on multi-agent was established. The simulation result shows that the intelligent forecast system provides a strong adaptability and the average forecast error of the system drop from 6.45% to 5.44% compared with single models.

Keywords: Multi-agent, forecast of electricity price, electric power market, fuzzy evaluation.

1. Introduction

Under the circumstance of electric power market, forecasting of electricity price is keenly concerned by all participators. In electric power market, if the short-term electricity price could be relatively accurately predicted, the production plan and the bidding strategy can be organized ahead in order to gain the maximum economic benefits. The short-term electricity price forecast (STEPF) means forecasting the crossing points between electric power supply curve and electric power demand curve several days earlier.

1.1. Influential factors and the analysis of difficulty of electricity price forecast

There are the 4 following aspects that influence the short-term electricity price:(1) Factors of the historical electricity price. The electricity price is strongly related to the same period of nearly several days, which follows the principle of "the nearer is bigger and the farther is smaller".(2) Factors of electricity consumption demand, including load demand, system backup capacity, etc. (3) Factors of electricity supply, including quotation strategy and quotation curve of power producer, available capacity of system, cost of fuel, etc. (4) Factors of period. The electricity price varies at different time periods in one day. For instance, the electricity price in the low load period is normally cheaper than in peak load period. (5) Policies on electricity price. When a new electricity price policy is carried into execution, the electricity price will change obviously. Though, the influence of the new price policy will weaken after a period of time.

Due to some man-made factors, it is difficult to forecast the electricity price: (1) The power producers adopt different quotation strategies; participators exert market power; conspiracy or gamble behaviors among the market, all these distort the real electricity price and make the forecast more difficult. (2) The electric power market is more of an oligopoly market, rather than a perfect competition market. There are only a few bidders in the market and the price is decided quite randomly. (3) The quotation strategy of power producer is regarded as commercial confidentiality, so analyzing and predicting the quotation behavior of power producer is also difficult. (4) The electricity power market is established not long ago, and not every power producer has installed stable quotation strategy and scheme.

1.2. Review of current methods of the STEPF

The STEPF methods fall into two categories: qualitative forecast and quantitative forecast. In the qualitative forecast, electricity price is predicted directly by experts. Qualitative forecast can make full use of all information. However, it might be too subjective, therefore different people would have different conclusions even based on the same information.

In the quantitative forecast, the electricity price is extrapolated by identifying the correlations among history data. Typical quantitative methods are:

Time series. This method assumes other factors as constant and the future price only relates with the history price. The model of recent price series is established firstly and then the future price is estimated. It applies to the situation in which the market is relatively stable and the influential factors change little. In application, the forecast accuracy does not reduce although its theory is relatively simple. [1,2,3,4]

Multiple linear regression. The principle is to determine the correlation between electricity price and influential factors according to the regression analysis of mathematics statistics, so as to forecast the price. [5]

Artificial neural networks. This method fits the relation between electricity price and influential factors by using artificial neural network's ability of approximating any non-linear functions. Up to now, the effectiveness of artificial neural networks, nevertheless, has not been proved by long-term continuous forecast. [6,7,8,9]

According to the above mentioned analysis: (1) there is no method absolutely better than any other. Every method has a certain limit of application and limit-exceeding, so that electricity price forecast in various market condition shall involve multiple methods comprehensively. (2) The time series method is suitable for the relative stable market. Both multiple linear regression method and artificial neural network are used to the market that fluctuates to some extent. If the market situation turns in, such as load demand changes dramatically, the above-mentioned quantitative methods will bring great errors. Therefore the expert's qualitative forecast has to be introduced. (3) As different forecast method is applied to different market fluctuation

conditions, it is necessary to assess the market fluctuation condition in order to select a proper method.

1.3. Basic ideas for improving the STEPF methods

Based on the above-mentioned analysis, we propose the following concepts to modify the STEPF methods:

1) Analysis of existing standard process of electricity price short-term forecast as well as its system structure.

2) Based on the above-mentioned analysis, further study on the STEPF architecture and forecast process shall be conducted in the following aspects:

①Hierarchical analysis of the "agent" architecture.

②Based on the "agent" hierarchical analysis, study on the composition of STEPF system based on multi-agent by using "IDEF1 method".

3) Based on the composition of "multi-agent" system architecture and "knowledge base", study on the STEPF procedure.

2. The analysis of the forecast process and the structure of the forecast system

2.1. Normal forecast process

Normally, the basic forecast process of the STEPF system is shown by Fig. 2.1:

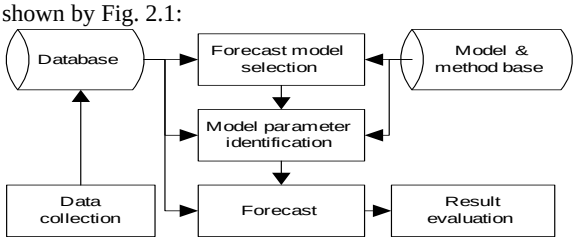


Fig. 2.1 the normal forecast process of the STEPF system

Notes to flow diagram:

1) Data collection: In this part, all types of information required by the electricity price forecast system is collected and stored into the database, including short-term load forecast data, system available capacity data, fuel cost information, etc.

2) Forecast model selection: In this part, one or more appropriate short-term electricity price forecast models are selected according to the current market condition.

3) Model parameter identification: In this part the parameters of the selected forecast model are identified.

4) Forecast: In this part the forecast is implemented by using the selected model. If there are several suitable models, a combination forecast should be conducted.

5) Result evaluation: In this part the multiple forecast results are synthesized and the forecast error is calculated according the real electricity price.

2.2. Overall structure and the analysis of the STEPF system

Based on the above normal forecast process, the "tree hierarchy model" is employed to depict the STEPF system, and its overall structure is shown by Fig. 2.2:

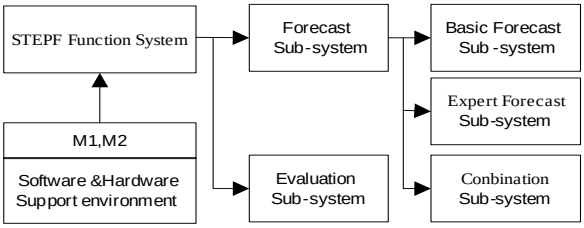


Fig. 2.2 the normal structure of the STEPF system

In the system shown by Fig. 2.2, M_i （I=1,2）denotes:

M_1: Database, storage all types of data required by electricity price forecast.

M_2: Model & method base, storage structures and parameters of time series, multiple linear regressions, artificial neural networks forecast models.

In STEPF, manual proactive participation is utmost important. In order to effectively support and control the "dynamic-characterized" system operation, issues like "man-machine interactive management", "communication management" between systems and "forecast process dynamic control" should be taken into consideration when constructing the system.

The normal information system struggles to meet the above-mentioned requirement, therefore it is necessary to adopt "multi-agent" architecture to design an "intelligent STEPF system".

3. Study on the STEPF system based on "multi-agent"

3.1. Architecture

According to the above analysis, the hierarchical "agent" architecture of the "intelligent electricity price short-term forecast system" is shown by Fig. 3.1.

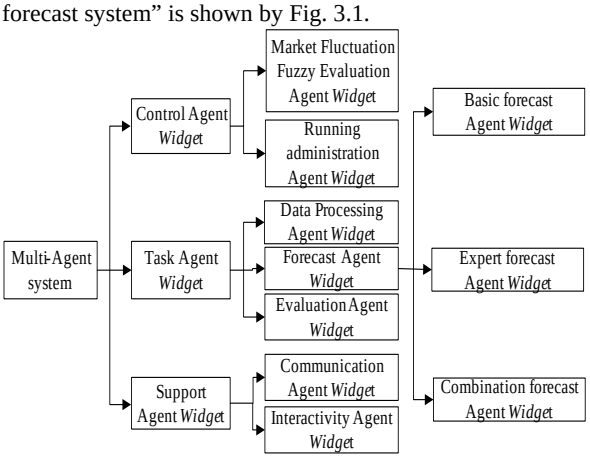


Fig. 3.1 the hierarchical architecture based on "agent"

Explanations of the "agents" are as follows:

1) Data Processing Agent Widget

Collecting all kinds of information required by electricity price forecast and checking the integrity and validity. Some invalid data will be filtrated and the lost data will be complemented according to the data of similar days. Then the data are stored into corresponding databases and can be transferred to every forecast agent directly.

2) Interactivity Agent Widget

This agent mainly composes the auto-generated information of the system and the information entered by hand. As to this issue, it synthesizes the call instruction

auto-generated by "Forecast Agent" and the call instruction entered by hand interactively.

3) Market Fluctuation Fuzzy Evaluation Agent Widget

This agent mainly deals with market fluctuation fuzzy evaluation by calling the Method 1 as specified in M2 (section 3.3), based on the information from the Data Processing Agent. It serves as the matching basis of calling relevant "facts" and "rules" described in the knowledge base (section 3.4) when the "Forecast Agent" is running by the support of the "Interactivity Agent".

4) Forecast Agent Widget

After the 3 types of "Basic Forecast Agent" and "Expert Forecast Agent" in M2 (based on the Method 2 in section 3.3), the "Combination Forecast Agent" is to run by adopting the Method 3 in M2, supported by the "call instruction" from the "Interactivity Agent".

5) Running Administration Agent Widget

This agent mainly dynamic controls the agents running in the system. Because good extension ability is required, the system should permit to add new agents and to stop some agents during its working process.

6) Communication Agent Widget

This agent has the standard function of managing the information communication among all "agents".

3.2. Overall architecture of the STEPF system based on "multi-agent"

Although the "tree model" architecture of forecast system shown by Fig. 2.2 has clear hierarchy, it fails to go deeply to disclose the logical relationship among functional systems. Therefore the US Air Force proposes a IDEF 1 Method to describe the system functions, so as to overcome the disadvantage of "tree model" in designing the information system.

The core part of this method is to design the system architecture as shown by Fig. 3.2, which is linked by 4 relations (input, output, administrate, mechanism), and then dissolve level by level according to the hierarchical order. Given that all 4 relations exist in the "multi-agent" system, this article will make use of IDEF 1 concept to define the overall logical model of the STEPF system based "multi-agent", which is shown by Fig. 3.3.

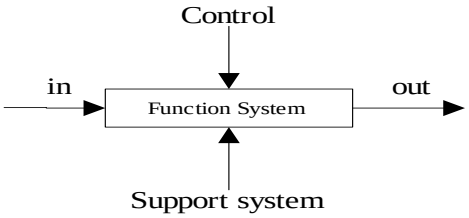
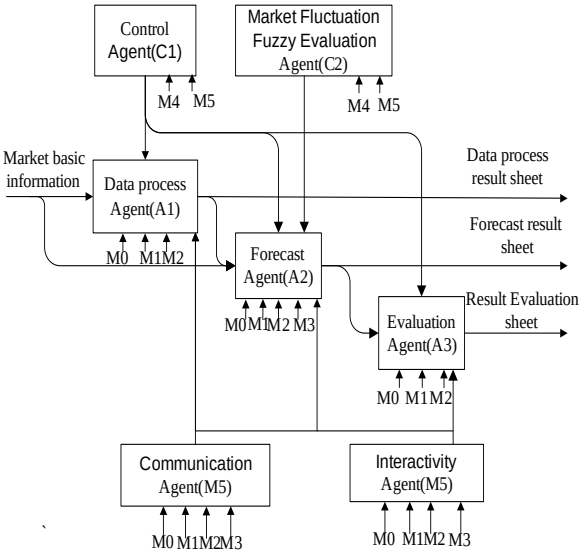


Fig. 3.2 the link structure of IDEF 1 Method



Notes: M0-forecaster, M1-Data base, M2-Model & Methed base,M3-Knowledge base

Fig. 3.3 the overall architecture of the STEPF system based "multi-agent"

3.3. Model Method Base (M2) Expansion

In the Model & Method Base shown by Fig. 2.1, 3 methods are added, which are abbreviated for Method 1, Method 2 and Method 3 respectively.

Method 1: Market fluctuation fuzzy evaluation

The power market operation experience shows that, if other influential factors remain constant, the electricity price also keeps still; if other influential factors changes drastically, then the price also changes drastically. According to this principle, the steps of market fluctuation fuzzy evaluation are:

1) Establishing factor set.

U={Load demand fluctuation range, system available capacity fluctuation range, system backup capacity fluctuation range, fuel price fluctuation range}

2) Establishing the evaluation set.

V={stable, low fluctuation, high fluctuation, drastic fluctuation}

3) Determining the individual factor evaluation matrix *R* and factor weight set *W* by using "Delphi Method".

4) Overall evaluation, fuzzy transposition.

$B=R*W$

The evaluation level corresponding to the maximum in *B* is the evaluation result.

Method 2: Expert forecast value composition

1) Setting x_i as the forecast result of i^{th} expert, *i*=1,2…n.

X as the composition expert forecast value.

$\bar{x} = \sum_{i=1}^{n} x_i / n$ as the average value of x_i.

$S = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 / (n-1)}$ as the standard deviation of x_i.

2) Then executing the follow circle:

i=0,k=0

From i=1 to n

If $\left|x_i - \overline{x}\right| > 3S$ *then* $x=x+x_i$, $k=k+1$

3) X=x/k

Method 3: Combination forecast algorithm based on self-adaptive weighted mean

Setting t as forecast period

x_{t-1} as the real value for $t-1^{th}$ period

$w_{t-1}^{(i)}$ as the forecast value of t^{th} period by adopting i^{th} basic forecast method

$w_{t-1}^{(4)}$ as the forecast value of t^{th} period by adopting expert forecast method

$\alpha^{(i)} = \left|w_{t-1}^{(i)} - x_{t-1}\right|$ as the variation between $w_{t-1}^{(i)}$ and the real value x_{t-1} of $t-1^{th}$ period

Then the combination forecast algorithm of self-adaptive weighted mean is described as follows:

$$\overline{w_t} = \sum_{i=1}^{4} \beta_i w_t^{(i)}$$

$\beta_i = \gamma_i(1/\alpha^{(i)})/(\sum_{i=1}^{4}\gamma_i\alpha_t^i)$ is the weight from the i^{th} method.

Here γ_i is the call instruction, $\gamma_i \in \{0,1\}$,i = 1,2,3,4,if $\gamma_i = 1$ then calling, otherwise not calling. This call instruction is generated based on the relevant "rule" in the knowledge base.

3.4. Knowledge base construction and the forecast procedures based on multi-agent

1）Knowledge base construction

For the short-term electricity price forecast based on multi-agent system, the knowledge base construction is very important. Below is some relevant knowledge that supports "forecast agent" within the knowledge base.

Fact knowledge

"Fact knowledge" refers to the knowledge that reflects the real market condition.

Five types of "fact" are defined:

Fact 1 Corresponding to the "stable" market condition, setting "status variable" *L* as 4.

Fact 2 Corresponding to the "low fluctuation" market condition, setting "status variable" *L* as 3.

Fact 3 Corresponding to the "high fluctuation" market condition, setting "status variable" *L* as 2.

Fact 4 Corresponding to the "dramatic fluctuation" market condition, setting "status variable" *L* as 1.

Fact 5 Corresponding to the "abnormal" market condition, set "status variable" *L* as 0.

The market condition is given by "Market Fluctuation Fuzzy Evaluation Agent".

Rule Knowledge

Corresponding to the "status variable" *L*, the rules to generate relevant "Forecast Agent" call instructions are as follows:

Rule 1 If *L*=4 then generate instruction "Agent-A211 and Agent-A212 must be entered".

Rule 2 If *L*=3 then generate instruction "Agent-A212 and Agent-A213 must be entered".

Rule 3 If *L*=2 then generate instruction "Agent-A213 and Agent-A22 must be entered".

Rule 4 If *L*=1 or *L*=0 then generate instruction "Agent-A22 must be entered".

Notes:

Agent-A211 - Time series agent

Agent-A212 - Multiple linear regression agent

Agent-A213 - Artificial neural networks agent

Agent-A22 - Expert forecast agent

2) The forecast procedure based on multi-agent

Supported by the "Knowledge Base (M3)", "Model Method Base (M2)" and "Database (M1)", the forecast procedure is shown by Fig. 3.4.

It is easy to get the main flow of this procedure from the diagram, i.e., supported by the "fact knowledge" in M3; the "Market Fluctuation Fuzzy Evaluation Agent" generates the evaluation value *L* (namely status value), which reflects the market fluctuation condition. Then according to the historical data in M1, extracting corresponding "rule" from M3 to generate compulsory "call instruction". The "call instruction" composes with the "manned call instruction" entered by hand based the 'Interactivity Agent". After that the "Running Administration Agent" administrates the forecast process, and accomplishes the STEPF in the "Forecast Agent". At the final stage, it outputs and evaluates the forecast results.

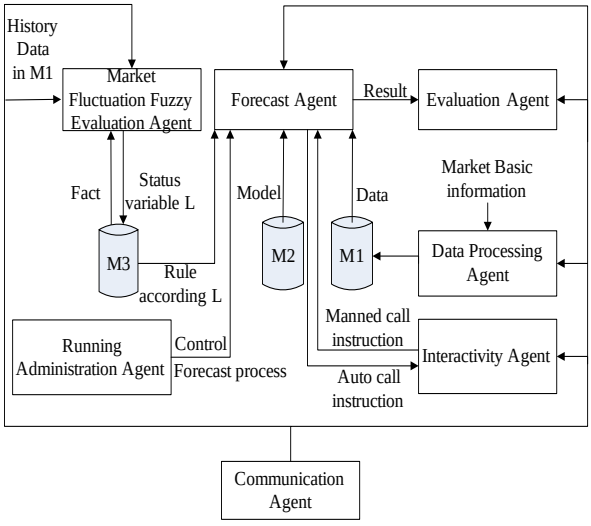


Fig. 3.4 the forecast process based on multi-gent

4. Forecast simulation

In order to verify the viability and effectiveness of the above design concept, we have utilized the real electricity price from 1 Apr.2003 – 30 Apr. 2003 in the power market in Zhejiang province and established a prototype system. The MAPE (Mean Absolute Percentage Error) of a day of 3 basic forecast models and the intelligent forecast system based multi-agent are shown by Fig. 4.1.

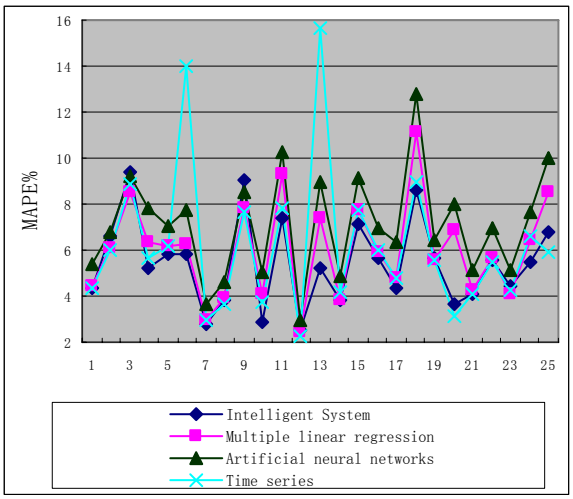


Fig. 4.1 the MAPE of 3 basic forecast models and the intelligent forecast system

The average forecast error of the system drop form 6.45% to 5.44% compared with single models. When the basic forecast models produce big error, however, the intelligent forecast system shows good forecast performance.

5. Conclusion

In the short-term electricity price forecast, none of forecasting method dominates the others obviously in all conditions and manual proactively participation is utmost important. AI technology will take an important role to build a forecasting system. The intelligent forecast system established in this paper can effectively enhance the adaptability and accuracy of the electricity price forecast. In the future, its forecast performance can be improved by means of increasing new basic forecast models and introducing more expert foreseeing experiences in knowledge base.

6. References

[1] X. Zhang, X. F. Wang, "The summarization of short-time electrictity price", Automation of Electric Power Systems, 2006, 30(3): 92-101.

[2] J. C. CUARESMA, HLOUSKOVA J, et al， "Forecasting Electrictity Spot-prices Using Linear Univariate Time-series Models. Applied", Energy, 2004,77(1): 87-106.

[3] J. C. CUARESMA, R. ESPINOLA, et al，"ARIMA Models to Predict Next-day Electricity Prices", IEEE Trans on Power Systems, 2003,18(3): 1014-1020.

[4] M. Zhou, Z. Yan, et al，"A novel ARIMA approach on electricity price forecasting with the improvement of predicted error", Proceedings of the Chinese Society for Electrical Engineering, 2004, 24(12): 63-68.

[5] Z.Y.Hu, W.Z. Sun, "Short-Term Medium-Term and Long-Term Forecasting of Electricity Price with Consideration to Market Power", Automation of Electric Power Systems, 2003, 27(22): 16-22.

[6] A. WANG, B. RAMSAY, "Prediction of System Marinal Price in the UK Power Pool Using Neural Networks", Proceeding of International Conference on Neural Networks.1997, 2116-2120.

[7] H. M. Yang, X. Z. Deng, "The analysis for the fuzzy characteristics of electric price and the research of its forecasting model", Power System Technology, 2004, 28(3): 59-64.

[8] A. HONG, C. HSIAO, "Locational Marginal Price Forecasting in Deregulated Electric Markets Using a Recurrent Neural Network In Proceeding and Confidence Interval Estimation with Cascaded Neural Networks", IEEE Trans on Power Systems. 2003,18(1): 99-105.

[9] L. Yang, M. X. Huang, "A Market Clearing Price Predictor Based on Modular Networks", Proceedings of the Chinese Society for Electrical Engineering, 2002, 22(8): 44-47.

Workflow Based Dynamic Roles Constraint Model for Coordinated Mobile Computing Environment

Jianyong PI, Xinsong Liu, Qingyun Fu, Ai Wu, Dan Liu
8010 Research Lab, School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan 610054, P. R. China
Email: pijy2004@163.com

ABSTRACT

Mobile computing systems introduce new requirements for role-based access control, which can not be met by using traditional RBAC. In this paper, a novel RBAC model of workflow based dynamic roles constraint relation is introduced to meet these requirements. This novel RBAC model formally describes the concurrent transaction logic by the extended predicate workflow model, and describes the dynamic constraint relations among the subtasks of workflows by analyzing the concurrent executive network of workflow. The static role constraint in the traditional RBAC was extended. The novel RBAC scheme is suitable for the workflow based mobile computing system. The analysis of the performance evaluating showed the novel RBAC model has favorable access control efficiency in the practical workflow based mobile computing environment.

Keywords: mobile computing systems, RBAC, concurrent transaction logic, extended predicate workflow model, dynamic constraint.

1. INTRODUCTION

Along the advance in wireless communication technologies, more and more powerful portable computing devices have been developed, which has made mobile computing a reality. The mobile computing environment requires neither the user to maintain a fixed and universally known position in the network, nor enable restricted mobility of the users. The mobile computing technology driving not only more and more functionality into mobile computing devices, but also security risk into mobile devices. The security in mobile computing environment is concerned more than before. Mobile clients roam between different networks and connect to different servers at unpredictable times. This poses great challenges to the resource access control in mobile computing.

In this paper, we proposed a logic framework to support coordinated RBAC in mobile computing environment. RBAC is now the research focus in the information security field. The RBAC model NIST (National Institute of Standards and Technology) proposed not only has perfect merit to users-oriented institutional framework but also is well better to the access control strategy such as least privilege and separation of duty [1]. It can realize the early access control strategy: DAC (Discretionary Access Control) and MAC (Mandatory Access Control) as well. Therefore it is provided with perfect application prospect. But as the information systems the people developed become more and more complicated, especially under the distributed mobile computing environment, the more dynamics and flexibility of the access control model are demanded [2,3]. Whereas the traditional RBAC is deficient to express the dynamic and flexible roles constraint. For example, in order to avoid making financial leaks, the accountant and the cashier in financial department are usually mutual exclusive. But in spare cash system, the two roles can be assigned to a person. According to this, the real roles constraint relations are not static but it can change dynamically with the changes of subtasks in workflows. So we introduce a new workflows model which is based on coordinated mobile computing via predicate description [4], discusses the dynamic roles constraint relations changing with the changes of subtasks in workflow among the system roles in the workflows system in mobile computing, and proposes the workflow based dynamic RBAC model which can describe the complicated constraint relations among the real roles flexibly and dynamically by the dynamic drive of subtasks in workflow. The workflow based dynamic RBAC model can fulfill any coordinated application demands under the mobile computing environment.

The rest of the paper is organized as follows. Section 2 introduces basic definitions of the predicate workflow model and the dynamic roles constraint in mobile computing. Section 3 introduces the lemma and proofs of the dynamic roles constraint. Section 4 presents workflow based dynamic RBAC model in mobile computing environment. Section 5 concludes the paper with remarks on future work.

2. DEFINITIONS OF THE PREDICATE WORKFLOW MODEL AND THE DYNAMIC ROLES CONSTRAINT FOR MOBILE COMPUTING

2.1 Correlative definitions of predicate workflow model based on concurrent transaction logic for mobile computing

In order to get the dynamic roles constraint relations in the mobile computing environment, we give the related definitions of the predicate workflow model firstly. Most workflows are described based on concurrent transaction logic, so we introduce some definitions of related characters of extended predicate. We postulate letter "*T*" as subtask set, letter "*R*" as roles set, letter "Φ" as empty set. We bypass some elements definitions in traditional RBAC model for having nothing to do with this paper.

Definition 1: workflow $W = \{t_1, t_2, ..., t_n\}$ is the set of subtasks. A workflow can be divided into many subtasks. Subtasks execute with concurrent transaction logic.

Definition 2: $\forall \alpha, \beta \in T, \alpha \otimes \beta$ is said that the relation between two subtasks is serial executive. That is, subtask β executes after subtask α.

This work is supported by the Sichuan Application Basic Research Project under Grant NO 04JY029-017-2.

Definition 3: $\forall \alpha, \beta \in T, \alpha \oplus \beta$ is said that the relation between two subtasks is parallel executive, i.e., the relation between subtasks α and β is "AND".

Definition 4: $\forall \alpha, \beta \in T, \alpha \vee \beta$ is said that the relation between two subtasks is optional executive, that is "OR".

Definition 5: $\forall \alpha \in T, \delta \subseteq Rand \delta \neq \phi, \alpha \wedge \delta$ is said that when subtask α execute, roles set δ which is subset of roles set R should be authorized and can not be empty. This indicates that when a subtask is executive, a role assigned is required at least.

Definition 6: $STRA: t \to 2^R$ is said the function image which is from subtask set to roles exponent set and *STRA(t)* is the roles set of subtask t. $RSTA: R \to 2^t$ is said the function mapping which is from roles set to subtask exponent set and *RSTA*（*r*）is the subtasks set of roles set r. A subtask may be authorized many roles when it's executive and a role may be authorized to many subtasks repeatedly.

Definition 7: $TRA: T \to 2^R$ is said the function mapping which is from tasks set to roles exponent set and *TRA*（*T*）is the activated roles set in the current workflow. A system might have many roles, and the roles set activated in an execution is only one of the roles sets in a system.

Each subtask executes with concurrent transaction logic in action, and there is a start node and an end node in which illustrated in Fig. 1.

The nodes in the Fig. 1. is said the temporary state of workflow. A temporary state can be inverted into another when subtasks are executive.

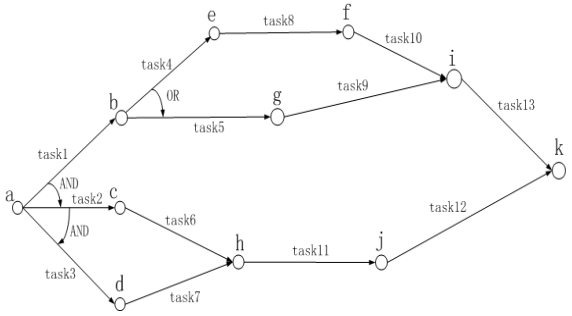


Fig. 1. The concurrent executive network of subtasks in workflow under mobile computing

In summary, the extended predicate logic defined above to Fig. 1. is follows:

$$a \otimes \left(\begin{array}{l} \left(task1 \otimes b \otimes \left(\begin{pmatrix} task4 \otimes e \otimes task8 \\ \otimes f \otimes task10 \end{pmatrix} \vee \begin{pmatrix} task5 \otimes g \otimes task9 \end{pmatrix} \right) \otimes i \right) \oplus \\ \left((task2 \otimes c \otimes task6) \oplus (task3 \otimes d \otimes task7) \right) \otimes \\ h \otimes task11 \otimes j \otimes task12 \end{array} \right) \otimes k$$

2.2 Definitions of dynamic constraint relations among roles

The subtasks in Fig. 1. can be simplified to roles set, which bypass other elements of the workflow. We can find that the constraint relations among the roles are complicated. But there are only four kinds of roles constraint relations in traditional RBAC model [5]:

For $assigned\_users(r: Roles) \to 2^{Users}$ the mapping of role *r* onto a set of users; $sessions = \{s_1, s_2, ..., s_n\}$ as the set of all sessions; $sessions\_roles(s: sessions) \to 2^{Roles}$ is the mapping of session *s* onto a set of roles.

1) Constraint of Static Separation of Duty

$$SSD \subseteq 2^{Roles} \times N,\ formally: \forall (rs, n) \in SSD,$$
$$\forall t \subseteq rs: |t| \geq n \Rightarrow \bigcap_{r \in t} assigned\_users(r) = \phi$$

That is, two roles can not be assigned to same user if they are assigned mutual exclusive constraint.

2) Constraint of Dynamic Separation of Duty

$$DSD \subseteq 2^{Roles} \times N,\ formally: \forall rs \in 2^{Roles}, \forall n \in N,$$
$$(rs, n) \in DSD \Rightarrow n \geq 2 \cap |rs| \geq nand$$
$$\forall s \in sessions, \forall rs \in 2^{Roles}, \forall role\_set \in 2^{Roles}, \forall n \in N$$
$$(rs, n) \in DSD, role\_set \subseteq rs,$$
$$role\_set \subseteq session\_roles(s) \Rightarrow |role\_set| < n$$

NIST had mutual description to Dynamic Separation of Duty , that is, we can assign many mutual exclusive roles to a user, and there is no conflict when several roles are activated independently. But when they are activated at the same time, there comes the conflict.

3) Order Constraint

If we must authorize a user with $role\_b$ before we authorize the user with $role\_a$, $role\_b$ is the fore-order constraint to $role\_a$ and $role\_a$ is the back-order constraint to $role\_b$.

4) Cardinality Constraint

If we limit the user numbers that a role can be authorized to, then there are constraints of up bound and down bound.

It's obvious that the constraint relations among roles of traditional RBAC model are difficult to express all kinds of constraint relations in real world. And it's lack of constraint logic definition maturity [6]. In order to express the dynamic and complicated constraint relations among roles, especially in concurrent and asynchronous distributed environment, i.e. distributed parallel computing environment, we must extend the static constraint relations among roles in traditional RBAC model. The following is the definitions of extended dynamic constraint relations among roles.

Definition 8: Given the roles sets activated by the concurrent subtasks execution path beginning with the same state jumping-off and assembling at the same terminal, this group of roles sets is said *concurrent constraint relations*. The roles sets in concurrent constraint relations can exit mutually, but the activating time must be limited by the assemble terminal. We use

Φ to denote the concurrent constraint relations among the roles sets.
Defined as:

$$\prod_{i<TRA(T)}^{\Phi} \{r \mid r \in roles_i, and\ roles_i \subseteq 2^R\}$$

Definition 9: Given the roles sets activated by the serial subtasks execution path are activated in order, this group of roles sets are *serial constraint relations*. The roles sets in serial constraint relations can not exit at one time, and the fore-order constraint roles sets must be back before activating the back-order constraint roles sets. We use Ψ to denote the serial constraint relations among the roles sets.
Defined as:

$$\prod_{i<TRA(T)}^{\Psi} \{r \mid r \in roles_i, and\ roles_i \subseteq 2^R\}$$

Definition 10: Given the roles sets activated by the optional subtask execution path having the relation to choose one from many subtasks, this group of roles sets is said *selection constraint relations*. The roles sets in selection constraint relation can not exit at one time. The roles sets in selection constraint relation determine the roles sets activated terminally depending on the last subtask. We use Ξ to denote the selection constraint relations among the roles sets.
Defined as:

$$\prod_{i<TRA(T)}^{\Xi} \{r \mid r \in roles_i, and\ roles_i \subseteq 2^R\}$$

Definition 11: Given authorizing a group of roles sets executed by subtask activates another group of roles sets or takes back all of the precondition constraints, the two groups of roles sets are *condition constraint relations*. This kind of relation can be regarded as a kind of complex relation: it's concurrent constraint relation among the fore-order roles sets, and so do the back-order roles sets. We use Ω to denote the condition constraint relation. N is the natural number set.
Defined as:

$$\prod_{i<TRA(T)}^{\Phi} \{r \mid r \in roles_i, and\ roles_i \subseteq 2^R\}\Omega$$

$$\prod_{j<TRA(T}^{\Phi} \{r \mid r \in roles_j, and\ roles_j \subseteq 2^R\}$$

3. LEMMAS OF EXTENDED DYNAMIC ROLES CONSTRAINT RELATIONS

According to the definitions of concurrent executive net figure which use predicate to describe subtasks, we can get the lemmas of extended dynamic constraint relations among roles Φ, Ψ, Ξ, Ω.

Lemma 1: Concurrent constraint relations Φ are anti-reflexive, symmetric, and transitive.

Proof: undoubtedly, it is no sense that concurrent constraint relations Φ are reflexive. For concurrent constraint relations Φ of roles sets have no precondition, it's symmetric. According to the Definition 8, the relations among roles sets in a same concurrent constraint relations set are concurrent by twos, so the transitive is doubtless.

Lemma 2: Serial constraint relations Ψ are anti-reflexive，anti-symmetric and transitive.

Proof: in the same way, it is no sense that concurrent constraint relations Ψ are reflexive. According to the Definition 9, the serial roles are activated by order. The execution to subtasks should not be reversible, so it is anti-symmetric. From the Definition 9, we can know that there is partial order in serial roles constraint relations, so it's transitive.

Lemma 3: Selection constraint relation Ξ are anti-reflexive，symmetric and transitive.

Proof: We can know that the selection constraint relations Ξ are similar to the concurrent constraint relations Φ. And the only difference is that each group of roles sets of the selection constraint relations Ξ is dynamic mutual exclusive. The proof progress is alike to the Lemma 1.

Lemma 4: Condition constraint relations Ω are anti-reflexive, anti-symmetric and transitive.

Proof: It is obvious the condition constraint relations are anti-reflexive. It is anti-symmetric for the condition relation. The relations Ω have partial order relations, so it is transitive.

4. WORKFLOW BASED DYNAMIC RBAC MODEL IN MOBILE COMPUTING ENVIRONMENT

We believe that the workflow based dynamic constraint relations among roles can express the complicated access control in real world more flexible, according to the discussion about the predicate workflow model and the extended dynamic constraint relations among roles. The workflow model based on concurrent transaction logic can truly express the dynamic states of the access between the access object and the system resources in actual actions. The access right to the system resources is presented by the system roles. But in workflow based dynamic RBAC model, the roles are bounded with dynamic subtasks to workflow. Therefore, the constraint relations among roles are decided by the execution, which avoids that the traditional RBAC model can't express complicated access control flexibly for the constraint relations among roles being static [7].

4.1 Subtasks Division

Concurrent subtasks in workflow division must abide by many principles, such as load balance、speed ratio、efficiency[8], and so on. To workflow based dynamic RBAC model, the concurrent workflow should be divided into subtasks with roles sets. There are some regulations below:

Regulation 1: $\forall \alpha \in T, \delta \subseteq Rand \delta \neq \phi, \alpha \wedge \delta$, i.e. it is required that not only subtasks being divided into one cannot be divided any more but also the subtasks having a

role at least when we divide a subtask. That's to say, a same role is needed by those subtasks being divided into, these subtasks should be combined into a subtask..

4.2 Roles division

The relations among the roles are successive and the roles are divided by the duties of the system in the traditional RBAC model [9]. So we must obey the following principles when we assign the roles of which are successive relations to subtasks.

Regulation 2: Given $RH \subseteq Roles \times Roles$ is partial order，marked as $\geq$. Roles is the roles set of a subtask, then

$$\{r_1 \mid r_1 \in roles \cap roles \subseteq STRA(t)\} \geq \{r_2 \mid r_2 \in roles \cap roles \subseteq STRA(t)\}$$

does not come into existence. It's obvious that the relations among the roles being successive in traditional RBAC model, i.e. partial order relation, only the son roles can be remained when the roles assigned to a same subtasks are successive for there are excessive privileges in father roles according to least privilege principle.

Regulation 3: There are static order constraint relations among roles in traditional RBAC, i.e. the precondition to authorize one role to the subtask is to obtain another role sets. It is probably a partial order relation in traditional RBAC, but in our workflow based dynamic RBAC model, we should assign the least role of partial order sets to the subtasks, i.e. assign the role of $MIN(r_1, r_2, ..., r_n \mid r_i \in (r, \geq))$ to current subtask. The static order constraint relations among roles can be depicted by extended dynamic constraint relations among roles, because the order constraint relations among roles is implication in the concurrent executive network.

4.3 The access control procedure

As the Fig. 1. shows, we depict the concurrent executive network of subtasks in workflow with predicate logic. Because the subtask is substitution of role set, so we omit the intermediate nodes in the paradigm. As follows:

$$\left(\begin{array}{l} \left(\begin{array}{l} roles_{task1} \otimes \left(\begin{array}{l} \left(\begin{array}{l} roles_{task4} \otimes roles_{task8} \\ \otimes roles_{task10} \end{array}\right) \vee \\ \left(roles_{task5} \otimes roles_{task9}\right) \end{array}\right) \oplus \\ \left(\left(roles_{task2} \otimes roles_{task6}\right) \oplus \left(roles_{task3} \otimes roles_{task7}\right)\right) \otimes \end{array}\right) \\ roles_{task11} \otimes roles_{task12} \end{array}\right)$$

According to the formula and Definition 8 ~ 11, we can get the following conclusion:

① $roles_{task4}$、$roles_{task8}$、$roles_{task10}$ are serial constraint relations, we can denote the relations among them as: $roles_{task4} \Psi roles_{task8} \Psi rols_{task10}$.

② $roles_{task5}$、$roles_{task9}$ are serial constraint relations, i.e. $roles_{task5} \Psi roles_{task9}$.

③ The role sets of ① and ② are selection constraint relations, i.e. $roles_{task4,8,10} \Xi roles_{task5,9}$

④ The role sets of ③ and $roles_{task1}$ are serial constraint relations ,we can denote the relations as: $roles_{task1} \Psi roles_{task4,8,10,5,9}$.

⑤ $roles_{task2}$、$roles_{task6}$ are serial constraint relations, i.e. $roles_{task2} \Psi roles_{task6}$.

⑥ $roles_{task3}$、$roles_{task7}$ are serial constraint relations , i.e. $roles_{task3} \Psi roles_{task7}$.

⑦The role sets of ⑤ and⑥ are concurrent constraint relations, i.e. $roles_{task2,6} \Phi roles_{task3,7}$.

⑧The role sets of ⑦ and $roles_{task11}$、$roles_{task12}$ are serial constraint relations. i.e. $roles_{task2,6,3,7} \Psi roles_{task11} \Psi rols_{task12}$.

⑨The role sets of ④ and ⑧ are concurrent constraint relations, we can denote the relations among them as: $roles_{task1,4,5,8,9,10} \Phi roles_{task2,3,6,7,11,12}$.

⑩ $roles_{task6,7}$、$roles_{task11}$ are condition constraint relations, i.e. $roles_{task6,7} \Omega roles_{task11}$.

Due to there is number of roles in every role sets, therefore the constraint relations in one workflow is complicated. The management for roles is dynamic in the workflow based dynamic RBAC model with predicate logic. We construct distributed management directory tree for constraint relations of roles with recursive algorithm in distributed mobile computing environment.

Algorithm:

Step 1: For P is the predicate logic to workflow, $T = \phi$ is initial empty directory tree, MAX is the maximum search degree.

Step 2: Analyze the constraint relations of subtasks in current bracket, and generate corresponding sub-tree of constraint relations, then construct T.

Step 3: Analyze the predicate logic in current bracket level from left to right, if the current bracket level is over, then judge if the current bracket level is boundary of predicate logic, if it's true then jump to **Step 4**, otherwise backtrack to up level. If the current bracket level is not over, then carry on, jump to **Step 2**.

Step 4: stop. The algorithm had generated directory tree for workflow based dynamic roles constraint in mobile computing.

The structure of directory tree is convenient for management to roles constraint relations, and implement with distributed LDAP (Lightweight Directory Access Protocol).

4.4 Simulation and Performance analysis

Based on the above formal framework, the workflow based dynamic RBAC model is implemented in mobile computing environment. Therefore the bottleneck of efficiency focus on the query procedure in distributed LDAP. In order to improve the efficiency of constraint relation query, we utilize the mechanism of distributed access vector cache, the effect is obvious. The experiment platform is based on the DP-Linux distributed parallel operating system, we have developed a mobile agent code, they can simulate mobile devices roam in different network servers. We evaluate the performance

with average request response time algorithm [10] in the platform, and make comparison with distributed SELinux. We assume the node server probability for request is μ, N is the PC quantity, B is the request arrival probability to system, so the average response time of server to request is:

$$t_m = C(\rho,\mu)/(\mu N - B) + 1/\mu$$

Where:

$$C(\rho,\mu) = \frac{(B/\mu)^N}{\left(\frac{B}{\mu}\right)^N + N!\left(1-\frac{B}{N_\mu}\right)\sum_{n=2}^{N-1}\frac{(B/\mu)^n}{n!}}$$

The concrete parameter of experiment platform is follows:

①Coordinated mobile agent code simulate the mobile devices roam in different network servers.

②DP-Linux distributed parallel operating system.

③PIII 800MHz CPU, 128M memory, 100M Ethernet adapter per server node.

④There are ten server nodes in DP-Linux;

⑤The load of test is reading and writing on files with 10M size randomly.

We postulate the security strategy is suitable in access procedure. Our scheme is denote as AB-RBAC (Agent-based RBAC), The illustration of experiment is showed as Fig. 2.:

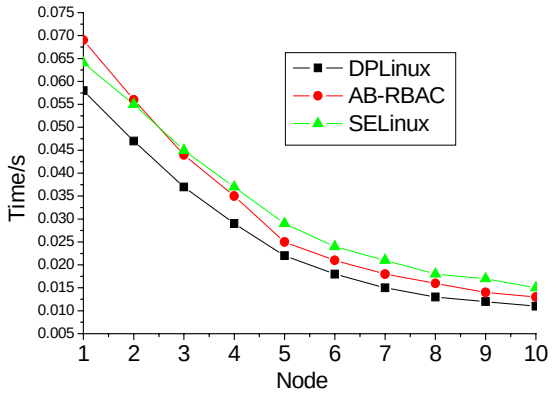


Fig. 2. Average request response time comp

According to Fig. 2., we can conclude that the more node quantity in mobile computing, the better performance of the mobile server cluster.

5. CONCLUSION AND FUTURE WORK

The workflow-based dynamic RBAC model for mobile computing analyzes the dynamic roles constraint with concurrent predicate workflow logic in distributed mobile computing environment, and the model extended the constraint relations in the traditional RBAC model simultaneously. The traditional RBAC model can not express the separation of duty and least privilege efficiently [11, 12], therefore we proposed the workflow based dynamic RBAC model, which can express the dynamic and complicated roles constraint relations in actual workflow system, i.e. distributed coordinated mobile computing environment in our paper, and the novel RBAC model combine the merit of the separation of duty and least privilege organically.

Because the complicated environment in mobile computing, such as wireless communication technology and package routing technology. The synchronization between mobile devices is difficult. Therefore constructing integrated access control mechanism for mobile computing environment is an attractive research field in the future. We will look into other implementation issues, such as temporal RBAC mechanism and spatio-temporal access control mechanism.

6. REFERENCES

[1] F. F. David, R. Sandhu, Proposed NIST standard for role-based access control, ACM Transactions on Information and System Security (TISSEC) 2001.

[2] B. Steinmuller, J. Safarik, Extending Role-Based Access Control Model With States, EUROCON'2001, International Conference on Trends in Communic-ations. ,Volume 2 , July 2001，4-7.

[3] F. Dridi, B. Muschall, et al, Administration of an RBAC system System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on 5-8 Jan. 2004, 187-192.

[4] H. Davulcu, M. Kifer, Logic based modeling and analysis of workflows, Proceedings of the seventeenth ACM SIGACT SIGMOD SIGART sympos- ium on Principles of database systems.

[5] R. Kuhn, Mutual exclusion of roles as a means of implementing separation of duty in RBAC system, Proceedings of the second ACM workshop on Role-based access control.

[6] W. B. Shim, S. Park, Toward an Improved RBAC Model for the Organic Organization, Ninth International Conference on Parallel and Distributed Systems, 2002, 17-20.

[7] M.J.Moyer, M.Abamad, Generalized Role-Based Access Control, 21st International Conference on Distributed Computing Systems, April 2001, 16-19.

[8] E. Bertino, E. Ferrari, et al, The specification and enforcement of authorization constraints in workflow management systems, ACM Transactions on Information and System Security (TISSEC).

[9] A.K.Mattas, I.K.Mavridis, et al, Towards Dynamically Administered Role-Based Access Control, 14th International Workshop on Database and Expert Systems Applications, Sept. 2003，1-5.

[10] X.S. Liu, The performance research of the distributed parallel server system with distributed parallel I/O interface. ACTA ELECTRONICA SINICA, 2002, (12), 1808-1810.

[11] A. Belokosztolszki, K. Moody, Meta-policies for distributed role-based access control systems, Third International Workshop on Policies for Distributed Systems and Networks, June 2002, 5-7.

[12] G. Kolaczek, Specification and verification of constraints in role based access control for Enterprise Security System, Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2003, June 2003. 9-11.

The Principal-agent Model Of Knowledge Sharing Within Enterprises

Peiqiang Tan
Information Management Department, Fudan University
Shanghai, 200433, China
Email: geffrey.tan@hotmail.com
Zhengchuan Xu, Hong Ling
Information Management Department, Fudan University
Shanghai, 200433, China
Email: zcxu@fudan.edu.cn
Email: hling@fudan.edu.cn

ABSTRACT

Nowadays, industry economy is evolving to knowledge economy. During the process, the core productive factor of enterprises is changing from labor force and capital to knowledge. More and more researches are being made on knowledge management in order to improve the core competence of enterprises. But on one side employees are reluctant to share their own knowledge with others, and on the other side enterprises lack of effective incentive mechanisms to encourage knowledge sharing. From the economic view the paper builds a principal-agent model of knowledge sharing within organizations, and then discusses qualifications that the incentive mechanism of enterprises should meet under two different conditions- one condition is employees' effort level for sharing knowledge is discrete, another is employees' effort level for sharing knowledge is continuous.

Keywords: knowledge sharing, principal-agent model, incentive compatibility

1. INTRODUCTION

At the age of knowledge economy, more and more enterprises are trying to improve the core competence through knowledge management. When implementing knowledge management, enterprises always want its employees to share their own knowledge with others and achieve the goal of knowledge sharing within organizations. However knowledge sharing with others is not spontaneous from the beginning because most of employees are reluctant to sharing their practical experiences accumulating through their hard working for many years, but enterprises encourage them to make contribution to knowledge sharing by releasing their personal experiences or knowledge [1]. In order to solve the conflict mentioned above, enterprises must design an effective incentive mechanism of knowledge sharing to enable its employees to create, share and apply knowledge. In September 2000, United Nation published one report named "Knowledge Management and Information Technology", which put forward incentives from leaders and management mechanism were two key factors affecting the successful implementation of knowledge management projects [2].

At present there have been several papers referred to the incentive mechanism of knowledge sharing. Meiyun Zuo made a structural research into the incentive mechanism of knowledge sharing from four parts, which were knowledge processing mechanism; knowledge explicit mechanism; knowledge performance mechanism and knowledge rewarding or punishing mechanism [1]. Jianhua Zhang and Zhongying Liu built measuring models for knowledge sharing, which included indexes system for assessing knowledge contribution of employees and evaluating methods for knowledge sharing status of employees. Furthermore they designed rewarding and punishing schemes for knowledge contribution status of employees [3]. Sulin Ba, Jan Stallaert and Andrew B. Whinston put forward knowledge sharing by way of setting up internal marketing mechanisms within enterprises [4]; Claude d. Aspremont, Sudinpto Bhattacharya and Louis_Andre Gerard-Varet discussed knowledge sharing mechanisms as public goods within enterprises [5]. Matthias Krakel analyzed factors affecting knowledge transferring between enterprises and employees [6]. However these papers just qualitatively analyzed the incentive mechanism of knowledge sharing within enterprises from the point of organization behavior or psychology; however they didn't quantificationally analyzed deeply. The paper analyzes knowledge sharing within enterprises from the economic point and builds a principal-agent model between enterprises and employees. Furthermore it discusses respectively two different conditions- one is concrete condition and another is continuous condition, and then concludes restrictions that the incentive mechanism of the enterprises should meet in the end.

2. THE PRINCIPAL-AGENT MODEL

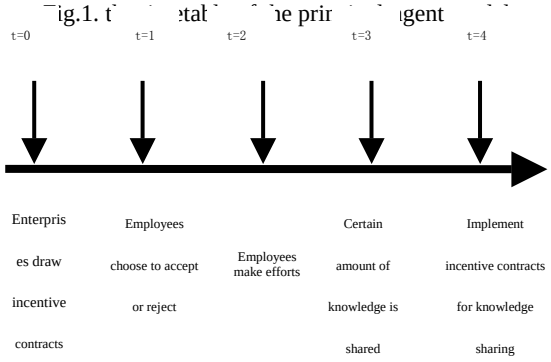
There exists an internal knowledge market within enterprises, in which there are six basic factors, such as buyer, seller, agent, price, principles and market failure [7]. Because of the existence of knowledge market, enterprises must set up an incentive mechanism conforming to economic principles in order to encourage employees to share their knowledge. The model discussed in this paper regards enterprise as the principal and employees as agents, whose knowledge contribution produced by his own efforts are uncertain and can't be inspected by enterprises. As a result enterprises have to pay employees just for the amount of knowledge, where we assume the amount of knowledge can be measured by various indexes [3]. In the following the paper discusses a simple principal-agent model under the condition that efforts and payoffs are both discrete, and then further discusses another model in which efforts and payoffs

* Supported by the National Natural Science Foundation of China (No.70572027)

are both continuous. Here we assume both enterprises and employees are risk neural.

The timetable of the principal-agent model is as Fig.1. At the first stage enterprises draw and deliver the incentive contracts for knowledge sharing; the second stage employees choose to accept or reject incentive contracts; the third stage employees make efforts; the forth stage certain amount of knowledge is shared; the fifth stage incentive contracts for knowledge sharing are implemented, employees receive payoffs and enterprises get the utility of knowledge sharing [8, 9].

Fig.1. t[illegible]etal [illegible]he prin[illegible]gent



2.1 Principal- agent model under the condition of the discrete effort level

The paper first discusses a simple model under the condition that the effort level of employees is discrete. We assume there are two kinds of effort level $e \in \{1,0\}$, and production is uncertain, so there are corresponding two kinds of states for the amount of sharing knowledge $K \in \{K_L, K_H\}$, the probability matrix is as follows (Table 1)

The cost of employees making efforts is $C(e)$ and $C(e=1)=e$, $C(e=0)=0$. Payoff for employees is $W(k)$ and utility of enterprises is $U(k)$.

We apply the contradictory method to solve the model mentioned before. At the stage t=2, when the effort level for

The amount of sharing knowledge (K)

Effort level (e)	K_L	K_H
e=1	$1-\Pi_1$	Π_1
e=0	$1-\Pi_0$	Π_0

knowledge sharing e=1, the employee's expected profit:

$$R_{e=1}=\Pi_1*(W(K_H)-e)+(1-\Pi_1)*(W(K_L)-e)$$

When the effort level for knowledge sharing e=0, the employee's expected profit:

$$R_{e=0}=\Pi_0*W(K_H)+(1-\Pi_0)W(K_L)$$

If we enable employees to make great efforts to share knowledge, it must be satisfied that the profit with efforts is more than the profit without any effort:

$$\Pi_0*W(K_H)+(1-\Pi_0)W(K_L)>\Pi_0*W(K_H)+(1-\Pi_0)W(K_L) \quad (1)$$

The inequation above is the incentive compatible restriction of knowledge sharing among employees.

when at the stage t=2 employees try their hard to share knowledge, the procedure turns to the stage t=1.The reserving utility is u if employees don't share their knowledge, so the qualification for employees accepting contracts of knowledge sharing is the expected profit of accepting the contract is more than his reserving utility:

$$\Pi_1*(W(K_H)-e)+(1-\Pi_1)*(W(K_L)-e)>u \quad (2)$$

The inequation above is the participation restriction for knowledge sharing among employees.

In the end enterprises make a choice at the stage t=0. Assuming enterprises set parameters in the model to enable employees to make efforts for sharing knowledge, so the net utility of enterprises

$$U=\Pi_1*(U(K_H)-W(K_H))+(1-\Pi_1)*(U(K_L)-W(K_L)).$$

The qualification for enterprises to implement contracts of knowledge sharing is his expected utility is more than his reserving utility, and we assume its reserving utility is 0:

$$\Pi_1*(U(K_H)-W(K_H))+(1-\Pi_1)*(U(K_L)-W(K_L))>0 \quad (3)$$

Contracts of knowledge sharing drew by enterprises have to satisfy (1), (2) and (3) restrictions mentioned above. Only under the condition, employees would like to make great efforts to sharing their knowledge with others.

2.2 Principal- agent model under the condition of the continuous effort level

Now the paper discusses further another model that the effort level of employees is continuous. The amount of knowledge sharing (K) is the random function of e. $K=K(e)$. The payoff for employees $W=W(K)$, the cost for efforts $C=C(e)$ and $C(e)>0$, $C''(e)>0$.So the profits function of employees $R=W-C=W(K)-C(e)=W(K(e))-C(e)$. If employees would like to share their knowledge with others, their profit when sharing knowledge must be more than their reserving utility:

$$W(K(e))-C(e)>u \quad (4)$$

The inequation above is the participation constraint of employees for knowledge sharing.

The utility function of enterprises is $U(k)$,and $U'(k)>0$, $U''(k)<0$. The net utility of enterprises

$$U=U(K)-W(K)=U(K(e))-W(K(e))$$

We maximize the net utility of enterprise as follows:

$$MaxU(K(e))-W(K(e))$$

Con

$$W(K(e))-C(e)=u \quad (5)$$

$$d_U/d_K * d_K/d_e - d_W/d_K * d_K/d_e = 0 \quad (6)$$

Enterprises want to pay less so the participation constraint for knowledge sharing is tightened, the remaining utility is owned by the enterprise. So the fifth constraint is equation. According to the model mentioned above we calculate the effort level of employees for knowledge sharing e^* when maximizing the utility of enterprises.

Besides we maximize the net utility of employees as follows:

$$MaxW(K(e))-C(e)$$

Con

$$W(K(e))-C(e)=u \quad (7)$$

$$d_W/d_K * d_K/d_e - d_C/d_e = 0 \quad (8)$$

According to the model mentioned above we calculate

the effort level of employees for knowledge sharing e^{**} when maximizing the utility of employees.

In order to make sure enterprises and employees have the same objectives, when the profit of employees is maximized, the utility of enterprises is maximized at the same time, e^{*} must be equal to e^{**}. $e^{*} = e^{**}$. Enterprises draw the payoff function of employees for sharing knowledge $W = W(K)$ which satisfies restrictions mentioned above, and enable behaviors of employees conforming to the benefit of enterprises.

3. CONCLUSION

During the process of implementing knowledge management, on one side employees are reluctant to sharing their knowledge with others, on the other side the incentive mechanisms for sharing knowledge within enterprises are imperfect, and so they can't encourage employees to make great efforts to share knowledge. As a result two aspects make a great influence on knowledge management within enterprises and even cause the failure of many projects. The paper builds the principal-agent model for sharing knowledge from the economic point and further discusses constraints the incentive mechanisms have to satisfy under the condition of two kinds of effort level- one is concrete and another is continuous. However the model built in the paper is still simple and needs to be improved further, for example we assume that employees are risk neural, but in fact some employees are risk preference or risk averse, so we should discuss respectively. The incentive mechanism for knowledge sharing within enterprises interrelates with many factors such as strategy, culture and trust between employees and so on. We should make careful consideration on factors affecting knowledge sharing within enterprises to build the incentive mechanism for promoting knowledge sharing and insure the implementation effect of knowledge management; eventually upgrade the core competence of enterprises.

4. REFERANCE

[1] M.Y. Zuo, "The incentive mechanism of enterprises for knowledge management", China &Foreign Corporation Culture, 2000, 24.

[2] X.F. Zhu, "The research of mechanism for knowledge sharing", Commercial Research, 2004, 22:112-116.

[3] J.H. Zhang, Z.Y. Liu. "Knowledge Contribution Inspiriting Mechanism for Knowledge Management", Journal of Tongji University (Natural Science), 2004, 32(7).

[4] S.L. Ba, J. Stallaert, et al, "Optimal Investment in Knowledge within firm using a market mechanism", Management Science, 2001, 47 (9): 1203-1219.

[5] Laude d. Aspremont, Sudinpto Bhattacharya, et al, "Knowledge as a public good: efficient sharing and incentives for development effort", Journal of Mathematical Economics 1998, 30:389-404.

[6] Matthias Krakel, "Withholding of knowledge in organizations", Schmalenbach Business Review 2002, 54:221-242.

[7] K.Y. You, "Knowledge Sharing and Enterprise Innovation", The Press of Tsinghua University, 2003.

[8] J. J. Laffont, D. Martimort, " The Theory of Incentive Ⅰ-The Principal-Agent Model", The Press of China Renmin University, 2002.

[9] S.Y. Xie, "Game Theory", the Press of Fudan University, 2002.

Research of Supply Chain Electronic Kanban Based On Mobile Agent

Yuan Li, Guangming Wang
Research Institute of Information Engineering and Electronic Commerce
ZheJiang GongShang University
HangZhou, 310035,China
Email: liyuan730@163.com **Email:** gmwang@hz.cn

ABSTRACT

As a high efficient production management way, The Just-in-time (JIT) is abroad used by enterprises. In the Supply Chain Environment, more enterprises unions regard this management philosophy as an efficient cooperate production way and accept it. The kanban is a critical part of a JIT system. In order to meet the requests of exchanging the kanban information among a union's members, a special information exchange tool is wanted. Mobile agent has characteristics such as autonomy, mobility and intelligent, so it's suitable to be used as the tool of exchanging kanban information. Discussing and studying the designation and realization about this kind of mobile agent is necessary and meaning.

Keywords: SCM, JIT, kanban, Mobile Agent.

INTRODUCTION

With the social developing and technological improving, the environment, which that's enterprises are facing, can be summarized as "customer-lead, intense competition, rapid changes". The competition of the modern market has changed from the competition among enterprise to the competition between enterprises union. Enhancing the competitiveness of enterprises union has become an urgent requirement. Supply Chain Management (SCM) formed in the context of increasingly fierce competition is a systematic management solution. By planning, coordinating and controlling information flow, logistics and capita, it focus on core businesses to make the suppliers, manufacturers, distributors, retailers, even the end-user into a network of functional chain[2]. An ideal supply chain is highly integrate, fluidly information exchange, and also can achieve the goal of customization. It's a dynamic enterprises union, which can farthest meet consumer demands.

1. THE REALIZATION OF JIT IDEAS IN SCM

1.1JIT and SCM

JIT stands for just-in-time, Its core idea is that producing a product on time according to the customer's need strictly, stimulating the production by demand, Making great effort to compress the inventories of the raw material, the semi-production and the production, eliminating waste, reducing costs and improving the efficiency. A principle, which can achieve this idea, is carrying the necessary materials, which have correct quantity and perfect quality to the necessary locations at the necessary time [4].

The JIT originally was a business-oriented internal production management solution, the effectiveness of carrying out the JIT heavily depends on the cooperation among businesses, and this cooperation restricts and impacts the results of implement the JIT by individual business[5,6]. The SCM based on the achievement of demand-production in the corporate alliance, the long-term and systematic strategic cooperation between the enterprises can provide a vaster of applied space for the implement of the JIT. For the supply chain management, Carrying out the JIT have great significance in improving response time for demand of the entire supply chain, reducing the costs of supply chain logistics and achieving the demand-supply in time.

1.2 The production information exchange under the supply chain environment

Kanban is a critical part of a JIT system. More specifically, a kanban is a card, labeled container, computer order, or other device used to signal that more products or parts are needed from the previous process step[3,7]. It contains information on the exact product or component specifications that are needed for the subsequent process step. Kanban works from upstream to downstream in the production process. At each step, only as many parts are withdrawn as the kanban instructs, helping ensure that only what is ordered is made, the upstream processes only produce what has been withdrawn [1].

Supply chain is a kind of expand system, its information flows differ from a single business enterprise. The traditional mode of JIT in business enterprise uses kanban as the tool of delivering production information between the processes. But in the supply chain management, the production unit extends from the internal individual process unit to the external enterprises, and there are spatial distances among these enterprises. Therefore, a kind of information exchange tool, which can deliver the information to the supplier, is indispensable. As figure 1:

At the same time, this information exchange tool, which can be used as kanban has to face the following problems:

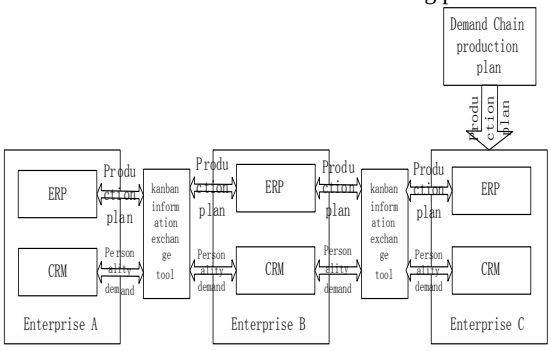


Fig. 1. production information deliver sketch

① EDI is always used as a traditional way of exchanging information among enterprises. The core manufacturing enterprises have to establish EDI communications with all its partners, this puts forward a higher request to the information system of the core business and its partners, and the cost of the information exchange is high. Therefore, this new information exchange tool must be easy to achieve, and have the feature of low-cost.

②Delivering information between enterprises will face the problem of systems heterogeneity, to enable the production information deliver to the upstream business timely and accurately, the heterogeneity have to be shielded. Therefore, this tool must be able to effectively overcome the problems of exchanging information between the heterogeneous systems and deliver the information accurately and transparently to the upstream enterprise.

③In order to ensure the quality of production and reduce cost, competition must be broached to the supply chain management. The relationship among the enterprise and its suppliers is a contractual relationship, enterprises constitute a dynamic of alliance, the node enterprise join and exit is inevitably, so it's requires that the information exchange tool must be easy to expand.

④The enterprises use the EDI as a traditional way to exchange information through the EDI appropriation net, it excessive reliance on the EDI center and this fault if the network breaks down, the production will face the interruption, and the same problem will inevitable if use the Internet as the transmission media. So the New information exchange tool must be able to overcome the problem such as network congestion and delay. It must have the characteristic of wisdom routing.

⑤ Achieving the JIT production between the enterprises in the supply chain, the transmitting of information between the upstream enterprises and the downstream enterprises is two-way and real-time. So this information exchange tool must have the automatic and intelligence feedback function.

2. REALIZE ELECTRONIC KANBAN BASE ON MOBILE AGENT

2.1 Electronic kanban designing based on the mobile agent

It's said that electronic kanban is an inter-enterprise production information exchange tool, and it includes the information of production, production time, production methods, production order, delivery volume, delivery time, location, tools and etc. Furthermore, electronic kanban will also include personalized requirements of downstream enterprises, such as qualities, specifications, delivery channels and delivery modes. Enterprises receive the electronic kanban, and then join it into their ERP information systems as a basis for developing their own production plans. At the same time in the process of production, their own production information will also be encapsulated into the information exchange tools, and feed back to the downstream enterprises timely.

This paper discusses that how to realize the information exchange tools use the mobile agent technology. Mobile Agent is a program, which can independently transfer from one host computer to another in heterogeneous network, and interact with other agents or other resources. It could package its own course code, data and control information into a message transmission transferring on the Internet. The cross-platform characteristic of mobile Agent can be used to overcome the heterogeneity between the enterprises' information systems and realize transparent information exchange for production[10,11]. The feature which mobile agent can interact with environment, can be used to achieve the kanban information automatic integrates into the ERP system and timely feedback of production information.

The mobile agent system will generally be divided into two parts: one part, which is called agent server, the other is called agent entity. The agent, which executes in the service facilities, uses ACL (Agent Communication Language) to communicate with each other and visit the services provided by service facilities [9].

The agent server makes agents transfer among hosts base on the agent transmission protocol. To coordinate and monitor the operation of agents and provide services interfaces for the agent, it establishes an operational environment and security protection mechanism for agents.

The agent server also has to exchange data with existing enterprise's information system through the ORB (Object Request Broker), which complies with the standard of CORBA (Common Object Request Broker Architecture). Each Agent Server loading its implementation code, data and operational status for each agent, and through this, an appropriate environment has been established [13]. As figure2:

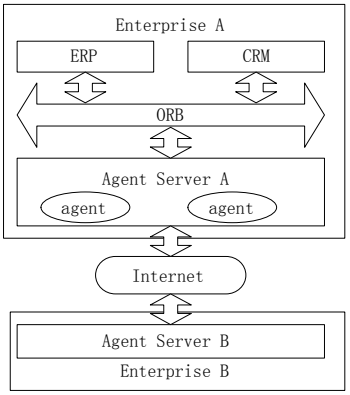


Fig. 2. working framework

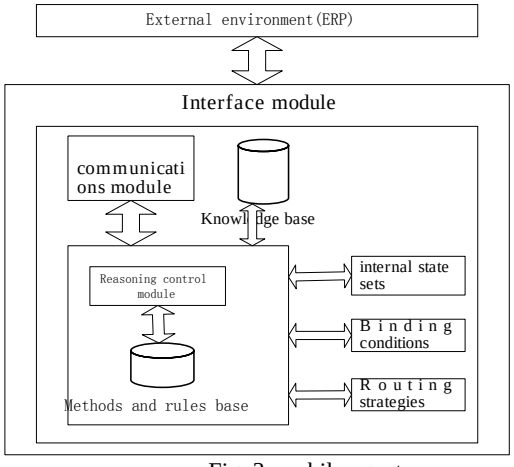


Fig. 3. mobile agent

Mobile agent works in the agent server and through the ACL to achieve the mutual communication and visits the service provided by the agent server[8]. The features of the mobile agent's structure as been shown in figure3:

The internal performance of an agent used as kanban is as following:

The communications module is responsible for the agent

communications between the mobile agent and the external environment (agent server). It realizes the ACL semantics receives the information from the external environment and sends the information to the designated agent or agent server. According to the information as well as their own request state, the mobile agent's reasoning and control module analysis, understand and reason, then implementation the actions, update their own state and save the information. This may lead to further activities such as creating sub-agents, timely feeding back information to the agent founder. The entire workflow as figure4:

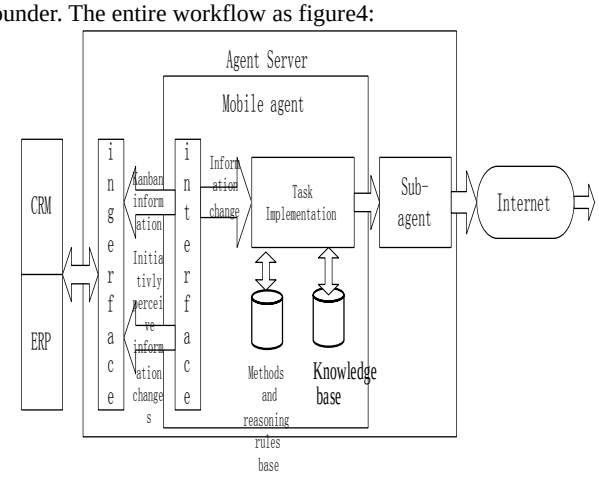


Fig. 4. mobile agent activities

The Interface module, also known as security agent module, is the intermediary of communication between the mobile agent and the environment. It carries out the security strategies and prevents the maliciously visit from the external environment. These measures can ensure the accuracy and the effectiveness of the kanban information.

Knowledge base is a perception world known by agent; it preserves reasoning methods and rules during the movement process.

The internal state sets is the current state of agent during its implementation process, it affects the agent's mission process.

Routing strategies decide the trails when agent mobiles, they are may be some static scheduling services, or rules-based dynamic routing which can meet the needs of complex and non-deterministic task.

Binding conditions are made by the agent founder to ensure the conduction and performance of agent, such as the time of return, the time of duration in a site, and so on.

The task-executing module includes the agent operation module, the reasoning methods and rules associated with the tasks.

Internal data module can save the processing data, which is required by the agent, and the collecting data, which will send back to the founders.

2.2 Realize an electronic kanban based on mobile agent

One feature of mobile agent is heterogeneous. Agent exists as series when it moves. In order to enable agent migrate in network conveniently and safely, so it have to adopt a cross-platform language. The Java language is applicable. Therefore it's commonly used to develop the mobile agent system. In order to overcome the drawbacks of the Java RMI mechanism and improve the compatibility and network transparency of the system, the Java language can be combined with the CORBA technology to develop a mobile agent platform[12].

and uses KQML to communications with other agents. It continuity. This electronic kanban has character of

3. CONCLUSION

In the supply chain environment, this thesis examines the realization of tools, which can exchange JIT production information among enterprises, aiming at the special requirements of transmitting information, and the exchange of kanban information by using mobile agent technology. This technology takes full advantage of mobile agent's characteristics such as autonomy, mobility, security and intelligent and nimble. It can dynamically participate in entrepreneurial activities so that enterprises in the league can develop faster.

4. REFERENCES

[1]Just-in-Time/Kanban, http://www.epa.gov/lean/thinking/kanban.htm.
[2] H.J.Yu, D.L.Yang, "SCM and Kanban solution", Forecast, 2003, 22(3).
[3] X.J.Yang,"Kanban solution and the competitiveness of Manufacturing enterprises", Journal of Hunan University, 2001, 15(4).
[4] G.C.Wang,"Study on innovative Tactics about SCM Based on JIT", ogistics technology, 2003,8.
[5] J.Li, H.C.Liu,"JIT Production Model and Use Tactics", Value engineering, 2002,3.
[6] Y.W.Zhang,"Explore of JIT", Inner Mongolia Science technology and Economy, 2004,12.
[7] F.W.Shan, B.G.Ceng,"Promoting and apply the Kanban solution in the in modern enterprise", Jiangxi Social Sciences, 2002,2.
[8] R.B.Wang, C.L.Zhou,"The Study of Mogent: An Overview", Application Research of Computers, 2001,6.
[9] Y.Y.Zhang, "The technology of mobile agent", public of tsinghua, 2003.
[10] Q.Xu,"Workflow and multi-agents Technology in SCM coordination management", Computer engineering, 2003,15.
[11] X.D.Chao,"The study of e-commerce agent technology", Microcomputer applications, 2000, 16(7).
[12] P.Bellavista,"Protection and Interoperability for Mobile Agents: A Secure and Open Programming Environment", IEICE Transactions on Communications. Special Issue on Autonomous Decentralized Systems, 2000, 83(5).
[13] P. Bellavista, A. Corradi, "CORBA Solutions for Interoperability in Mobile Agent Environments", Proceeding of the International Symposium on Distributed Objects and Applications (DOA'00), 2000.

Design and Implementation of Search and Rescue At Sea Decision Support System Based On Multi Agent

Weihong Yu
Economics and management school, Dalian Maritime University
Dalian, Liaoning 116026, China
Email: yuwhlx@163.com

And

Chuanying Jia
Navigation school, Dalian Maritime University
Dalian, Liaoning 116026, China
Email: chuanyingjia@sina.com

ABSTRACT

It is very important to make a search and rescue plan promptly and efficiently after the perils of the sea accident occurred. The advanced Agent technologies has been applied into the area of search and rescue at sea decision in this paper, and the architecture of search and rescue at sea decision support system based on multi agent has been proposed. Conversation models among agents have been established and the conversation models' validity has been verified by using Petri Net. As a new development method, JAFMAS has been applied into the implementation of multi agent system. The result indicates that compared with the traditional software system, search and rescue at sea decision support system based on MAS fuses the agent technology, neural network, data mining, ontology and other advanced technology. It can provide a distributed, isomerism, coordinated development environment.

Keywords: JAFMAS; Multi Agent; Search and Rescue at Sea; Decision Support System; Petri Net; Conversation Model.

1. INTRODUCTION

At present, the frequent perils of the sea accident not only caused the great losses on politics, economy, and military all over the world but also polluted the marine environment. It is extremely significant to complete search and rescue effectively.

In order to promote the information construction of the search and rescue work and to help RCC make an effective plan, we attempt to develop the advanced search and rescue at sea intelligent decision support system. In recent years, Theoretically said, search and rescue at sea needs the various coordination and the cooperation, needs to process the massive multiple source information, without doubt, it is a good test bed for carrying on Agent and MAS research.

There are some workable platforms for developing MAS such as JATLite, Aglets, and Jade and so on. Compared to other platforms, JAFMAS uses Java features to provide a framework that not only guarantees essential communication features between agents but also provides coordination mechanisms to application developers [1]. It does not adhere itself to any centralized concept and provides for developing scalable, fault-tolerant, self-configurable and flexible multiagent systems. Therefore our system mainly uses the JAFMAS from analysis to the implementation.

2. BRIEF INTRODUCTION TO JAFMAS

The Java-based Agent Framework for Multiagent Systems (JAFMAS) provides a generic methodology for developing speech-act based multiagent systems, agent architecture, and a set of classes to support implementing these agents in Java.

JAFMAS provides communication, linguistic and coordination support through sixteen Java classes. Communication support is provided for both directed communication and subject-based broadcast communication [2]. This feature enables the user to develop scalable, fault-tolerant, self-configurable and flexible multiagent systems. Linguistic support is provided for speech-act (e.g. KQML) based languages allowing for agent independent semantics.

In JAFMAS, a five-step process is presented for implementing a multiagent system: (I) identifying the agents, (II) identifying the agent conversations, (III) identifying the conversation rules, (IV) analyzing the conversation model, and (V) MAS implementation. Here JAFMAS classes are extended for each agent, for each conversation, for each conversation rule, and for an operator interface to develop application specific classes.

These five steps are fulfilled by us in search and rescue at sea intelligent support system.

3. DESIGN AND IMPLEMENTATION OF SEARCH AND RESCUE AT SEA DECISION SUPPORT SYSTEM BASED ON MULTI AGENT

3.1 Identify the Agents of Search and Rescue Area Subheadings

This system is aimed to aid RCC to make search and rescue plan. After studying a lot of search and rescue cases, six tasks have been decomposed for making search and rescue plan according to some experts' experience [3]. Different Agents implement these six tasks.

Agents that have the similar aims and services are classified into one group. According to the tasks of the search and rescue and the agent's services, we identified four agent classes: user interface class, resource allocation agent class, search calculation agent class and search and rescue coordination agent class.

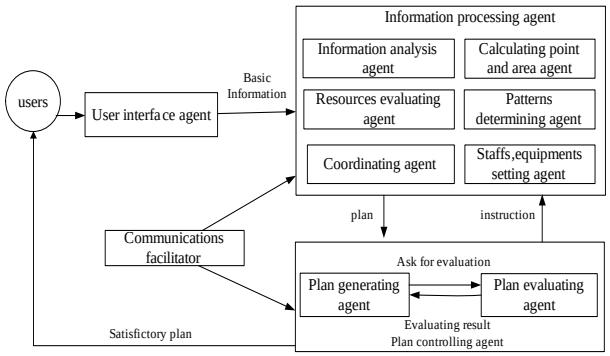


Fig. 1. the architecture of search and rescue decision support system based on MAS

After processing the basic information (this work is done by information processing agent which includes information analysis agent, resource evaluating agent, coordinating with others agent and calculating point and area agent), the first plan will be generated by plan generating agent. A satisfactory plan, which has been evaluated by plan evaluating agent, will return to the users. If the plan is unsatisfactory the plan-evaluating agent will ask for the information-processing agent to make a new plan all over again.

In a multiagent system the problem needn't to be solved step by step. Any agent will accomplish its functions if the available information is abundant.

Fig. 1. shows the cooperation among the agents.

3.2 Identify the Agent Conversations

In this step, we begin to model the interaction between the agents in the form of conversation. Finite State Machine is used to do that. Every possible conversation that each agent can engage in is identified. This method is demonstrated through the example application of establishing conversation model for user interface agent.

The Interface Agent is primarily assigned to search other agents on the net and to gather computation results. It provides the user with a control panel through which he can be informed about the available agents on the net and the resources provided by them. In this system the functions of user interface agent include:

1) Accept the input information. The input information includes the shipwreck data, the data about computing the drift and the weather information, etc.

2) Asset the situation according to the input data, define the emergency and the type of the shipwreck. Some decisions also can be made by the user interface agent.

These decisions include:

- What search and rescue resources should be dispatched;
- Whether need to calculate search area or not;
- Whether need the medical service aids;
- Whether need the corporation of other search and rescue organizations.

After making the above decisions, the user interface agent sends the task request to other agent. During the course of the execution, if one agent needs other agent's result, it can query the user interface agent to get the result. According to the above analysis, the finite state machine representation of the user interface agent's conversation is shown in Fig. 2..

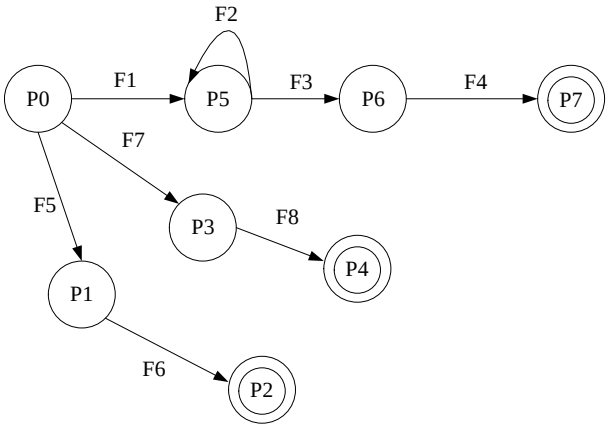


Fig. 2. The user interface agent's conversation model

3.3 Identify the Conversation Rules

Conversations are rule-based description of what an agent does in a certain situation. These rules define each transition of the finite state machine. In Fig. 2., F1 to F8 are all conversation rules. The description of the conversation rules is described in Table 1.

Table 1. The description of the conversation rules

Rule	Rule's description
F1	Transmit: propose a message containing the parameters for processing to the asset allocation agent.
F2	Received: receive a "ask for an answer" message from the assess asset allocation agent. transmit: reject to the asset allocation agent.suchthat: no answer now.
F3	transmit: propose answer to the asset allocation agent.
F4	Received: receive "done" message from the asset allocation agent.
F5	Transmit: propose a message containing the parameters for processing to the search calculate agent.
F6	Received: receive "done" message from the search calculate agent.
F7	Transmit: propose a message containing the parameters for processing to the coordinate agent.
F8	Received: receive "done" message from the search calculate agent.

3.4 Analyze the Conversation Model

In order to design a multiagent system that finds a coherent solution to the entire system problem, it is important to do an analysis of the logical consistency of all agent conversations. They should be analyzed to verify the coherency of the system. If any inconsistency exists in the model, the design is returned to step 2 of the methodology for redefinition of the conversations.

Petri Net models have emerged as a very promising performance modeling tool for systems that exhibit concurrency, synchronization, and randomness [4].This paper applies Petri Net to analyze coherency and

coordination of agent conversations.

A Petri Net is a four-tuple (P, T, IN, OUT) where

P = {p1, p2, p3... pn} is a set of places;

T = {t1, t2, t3... tm} is a set of transition;

$P \cup T \neq F$, $P \cap T = \Phi$;

IN: (T X P) $\rightarrow$ N is an *input function* that defines directed arcs from places to transitions, and

OUT: (T X P) $\rightarrow$ N is an *output function* that defines directed arcs from transitions to places.

The Petri Net representation of all conversations is shown in Fig. 3.In order to verify the correctness, we simplify it and design an incidence matrix. From the initial state of Petri Net M0, through computing of the incidence matrix, we can get all reachable states of M0 and draw out its state matrix.

The incidence matrix is $W = OUT\text{-}IN = w_{ij}$ where

IN: $T X P \rightarrow \{0, k\}$ is the input incidence matrix

OUT: $T X P \rightarrow \{0, k\}$ is the output incidence matrix

Here, $k \geq 1$ is an integer and denotes the weight of the associated arc. $IN(t_i, p_j)$ is the weight of the arc $p_i \rightarrow t_i$. $OUT(t_i, p_j)$ Is the weight of the arc $t_i \rightarrow p_j$? This weight is *0* if the arc does not exist. Row *i* of this matrix corresponds to the marking modification made by the firing of transition t_i.

The incidence matrix is used to find out the change in the marking of a Petri Net upon firing a given transition and the characteristic equation of the change in the marking is given by $M' = M_0 + T.W$, where *T* is the firing vector stating which transition has fired, M_0 is the initial marking, M' is the modified marking, and *W* is the incidence matrix.

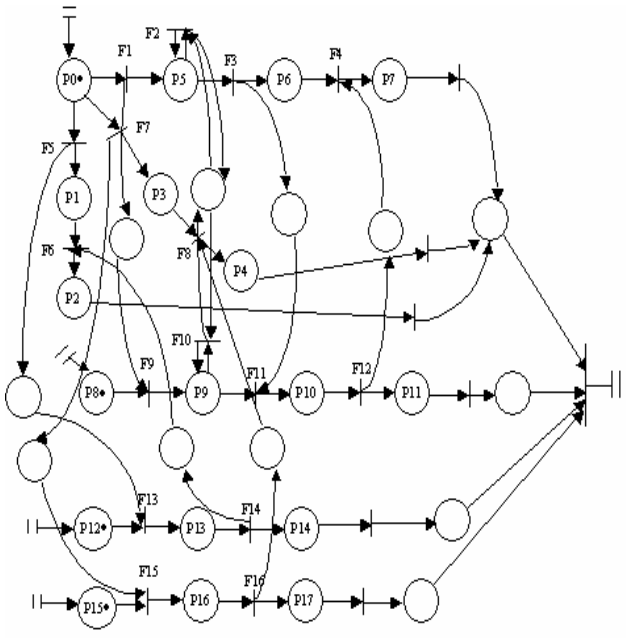


Fig. 3. the petri net representation of all conversations

After computing, it can be seen that in all possible marking of the Petri Net, the number of tokens at any place never exceeds one. Thus the Petri Net is safe [5]. We also get the reachability graph of the system. Each node of this graph represents a reachable marking of the Petri Net. We can see from this graph that from all the reachable markings of the system, a firing sequence exists, which contains every transition of the Petri Net. That implies that every transition of the Petri Net is live and the system is deadlock free.

So we can draw a conclusion that the original conversation models of the agents are right.

3.5 MAS implementation

MAS' implementation relates to some important classes: agent class, conversation class, conversation rule class, message class, basic input class and result data class. Another key point is database interface.

The agent class extends the Thread class of Java, so each agent has its own thread of execution. Thus, different agents which have been created within the same program space can run concurrently. In order to communication with others, each agent can bind with host computer by making use of RMI technology. Messages are dealt with through conversation. Each Agent has its own interface on which agent's attributes, events, message and current conversation are shown. In MASAR system we give a definition of each agent as follows:

masar_agent (attributes knowledgebase, methods, and interface)

Among those, masar_agent expresses the name of each agent, the name is unique.

Attributes: a set of all attributes such as the name, address, capability of an agent.

Knowledgebase: stands for the knowledge bases of each agent, which store the methods to solve some kinds of problems.

Methods: a set of methods, that is, the actions which an agent will take.

Interface: An interface through which an agent and the environment can have an effect on each other.

Conversation class also extends the Thread class of Java. This class overrides the run () method of the Thread class in order to continuously run its rule interpreter function. The main function of the conversation is to deal with messages according to conversation rules, and most functions of the Agent are fulfilled through conversations. Each conversation has one initial state and many final states. As soon as an agent starts a conversation, it adds the conversation instance to its conversation list. When the conversation reaches one of its final states, it remains sleep state and suspends the conversation execution. At this time, the agent can remove this conversation from the conversation list and also can switch it to the initial state or any other state to continue to work.

Conversations are rule-based descriptions of what an agent does in certain situations. Each conversation rule describes the actions that can be performed when a conversation is in a given state. Each rule has a current state in which it starts and a next state, which it reaches upon execution. If the present conversation state is the same as the current state of the rule and if the conversation is able to execute the rule then the next state variable of the rule sets the present conversation state.

The Message class implements Java's Serializable interface and can be transported using RMI.According to the rule of KQML, the message is defined as KQML primitive, and the message class has the following attributes: type, intent, content, the name of the conversation, sender, receiver and the time of send and receive.

The message class is described as following:

Class Message implements Serializable

```
{
private String type=null;
private Object sender=null;
private Object receiver=null;
private Object content=null;
private String intent=null;
private String convName=null;
}
```

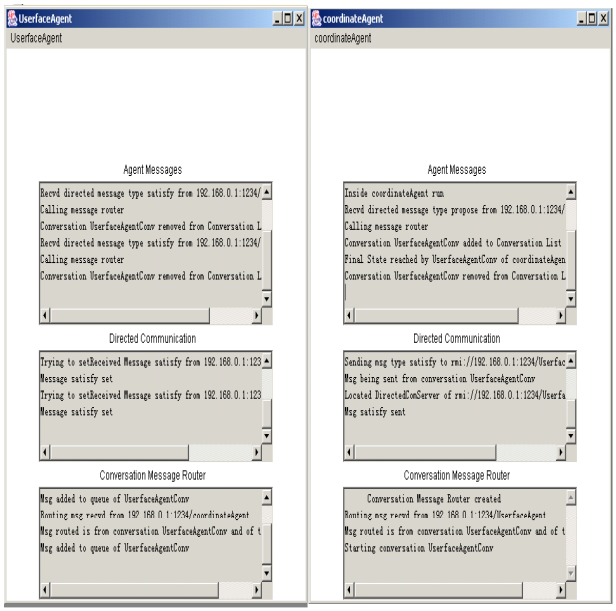


Fig. 4. Agent's running state

SQL Server dbms is used in this system. Some tables are created to store data such as the agent's basic information, the information of search and rescue's resource and so on. We adopt JDBC-ODBC as a database interface to visit SQL Server database.

When the system starts, the user can input the data about shipwreck, the user interface agent sends these data to other agents and starts conversation with others. The agent that processes the data can send the result back the result to the user interface agent. After the user interface agent receives the result, the conversation ends. The running state of the Agent is shown in Fig. 4.

4. CONCLUSIONS

We chose certain groups of representative data to carry on the simulation of the system. The result indicates that compared with the traditional software system, search and rescue at sea decision support system based on mas fuses the agent technology, neural network, data mining, ontology and other advanced technology. It can provide a distributed, isomerism, coordinated development environment.

Its advantages are:

1) Distribution: This system can provide the highly effective data exchange method for the experts of different geography positions.

2) Isomerism: Platform Independent customer supports are provided by this system.

3) Openness: The new technical modules and the function modules can be embedded into the system easily and the system does not need to re-construct.

4) Effectiveness: It shifts the information interactive load to each Agent, thus reduces the coupling degree between the system and the complex degree of subsystems.

5) Stability: Agent has the independency, thus the entire system stability can be guaranteed. So the entire system cannot collapse even if one subsystem fails down.

5. REFERENCES

[1] Deepika, Chauhan, JAFMAS: "a multiagent application development system", International Conference on Autonomous Agents Proceedings of the second international conference on Autonomous agents, 1998, 100-107.

[2] A. K.Galan, A. D.Baker, "Multi-Agent Communication in JAFMAS", In: Proceeding of the Workshop on Specifying and Implementing Conversation Policies, Third Annual Conference on Autonomous Agents, Seattle, WA, May 1, 1999, pp.67-70.

[3] W.H.Yu, C.Y.Jia, "Search and Rescue At Sea Decision Supporting System Based on Multiagent", Proceedings of 2003 International Conference on Management Science & Engineering, 2003.

[4] R. Hamadi, B. Benatallah, "A Petri net-based model for web service composition", Conferences in Research and Practice in Information Technology Series Proceedings of the Fourteenth Australasian database conference on Database technologies 2003,17,191-200.

[5] N.A.Anisimov, E. A. Golenkov, et al, "Compositional Petri Net Approach to the Development of Concurrent and Distributed Systems", Programming and Computing Software, 2001, 27: 309-319.

An Intelligent Agent platform applied in Internet Commerce

Qiumei Pu, Qianxing Xiong
School of Computer Science and Technology, Wuhan University of Technology
China, Hubei Province, 430063
Email: xqx@mail.whut.edu.cn
Email: puqm@mail.whut.edu.cn

ABSTRACT

With the increasing importance of electronic commerce across the Internet the need for agents to support both customers and suppliers in buying and selling goods or services is growing rapidly. It is a great challenge to build the software platform to support dynamic electronic commerce. This paper presents an electronic payment system based on agents. The aim of it is to provide a solution to automating trading process in the electronic marketplace. This paper describes one of the implementations of the intelligent agent in Internet commerce. In this paper, the process flow and the important part of the electronic payment system is analyzed. The proposed system model in this paper provides a new solution to the development of intelligent electronic commerce applications. The prototype system will implement the intelligence of the e-commerce with software agent, such as buyer agent, commodity agent, merchant agent and delivery agent.

Keywords: Electronic commerce, Electronic payment, Intelligent Agent, Ontology.

1. INTRODUCTION

With the increasing importance of electronic commerce across the Internet the need for agents to support both customers and suppliers in buying and selling goods or services is growing rapidly. Today there are several Web sites where people can look for a specific good or service but most of them do not take advantage of the intelligent systems and knowledge technology to automate the process and few help buyers and sellers. The traditional electronic commerce systems on the Internet require a large effort from users and include searching for parties interested in selling or buying what users want to buy or sell, comparing prices and other features of the good or service in order to make an optimal purchase decision. Neither dynamic price according to the actual demand nor customer specific offers is possible. The customer and the supplier cannot negotiate the details of a deal.

The traditional electronic commerce approach requires a direct communication between the potential buyer and seller. The buyer collects information about a product by directly accessing the information provided by the seller. If the offer satisfies his needs he negotiates directly with the seller of the product. Negotiation, delivery and payment are usually done manually. As a possible solution, an intelligent agent is a self-contained, autonomous software module that could perform certain task on behalf of its uses; it can assist the customers with this task. In order to enable Internet commerce transaction, the specific stages of business process must be expedited with a greater degree of automation on both the buyer's and the seller's side. The present work will focus on developing the mechanisms to automate some processes related to consumer's buying behavior and merchant's selling behavior model in the electronic market. How to develop the own safe electronic payment system of e-commerce is the key to accelerating the development of e-commerce on the Internet.

The remainder of this paper is organized as follows. In the next section we give a brief description of some applied agent-mediated electronic commerce system and ontology in e-commerce. In Section 3, we provide an overview of the complete architecture of the electronic commerce system based on mobile agent, followed by a detailed discussion of the main workflow and some secure measure to this model. Finally conclusions are drawn and future work is addressed in Section 4.

2. E-COMMERCE

Electronic commerce is already the fastest growing area in the superheated new Internet economy and carries potentials beyond measure. This is happening for two reasons. First of all, electronic commerce is extending business models. It reduces costs and extends existing distribution channels and may even introduce new distribution possibilities. Secondly, it enables completely new business models or gives them a much greater importance than they had before.

2.1 AGENT IN E-COMMERCE

Mobile agent architecture can bring many potential benefits such as great flexibility and improved performance into distributed systems. Therefore, they are believed to be playing an important role in future electronic commerce system. Not only can they provide a very flexible approach for information gathering on prices and goods available from the several catalog servers they visit, but also they can electively take over the different aspects of the electronic transaction, from price settlement to paying and delivery of the goods purchased.

Software agent [1] technologies provide a brand-new scenario that is used to develop the new-generation e-commerce system, in which the most time-consuming stages of the consumer's buying process will be automated. An electronic commerce system more or less includes six stages [2]: need identification, and product brokering, merchant brokering, negotiation, purchase and delivery, and product service and evaluation. Some agent-mediated prototype systems have been developed to automate some of these stages.

IT Media Lab's Kasbah [3] is an agent-based market for buying and selling goods. A user wanting to buy or sell a good creates an agent, gives it some strategic direction, and sends it off into a centralized agent marketplace. Kasbah agents can perform negotiation and settlement of deals according to the user's choice. This system is a web-based multi-agent classified system where users create buying and selling agents to help transact products. These agents automate much of the merchant brokering stage and

negotiation stage for both buyers and sellers.

Tete-a-Tete [2] is an agent-mediated shopping system, its agents cooperatively negotiate across multiple terms of a transaction. This negotiation takes the form of multi-agent, bilateral bargaining. It uses integrative negotiation interaction model and integrates all three of the product brokering, merchant brokering, and negotiation stages.

The Minnesota AGent Marketplace Architecture (MAGMA) [4] is a prototype for an agent based virtual market that includes all elements required for simulating a real market. These elements include a communication infrastructure mechanism for storage and transfer of goods, banking and monetary transactions, and economic mechanisms for direct or brokered producer-consumer transactions. MAGMA is an extensible architecture that provides all services essential to agent based commercial activities. This system supports the stages of need brokering, product brokering, merchant brokering, purchase and delivery.

Similar to the above systems, the architecture we propose for a marketplace includes the infrastructure required for conducting commerce on the Internet, supports communication among agents and allows for various forms of automated and human controlled transactions. In our research, we propose a new architecture for e-commerce systems. We present a platform that uses intelligent agent technology for dynamic e-business.

2.2 ONTOLOGY IN E-COMMERCE

Ontology may provide support in integrating heterogeneous and distributed information sources about a knowledge domain. Thus, ontology plays an important role in areas such as knowledge management and electronic commerce. Electronic marketplaces are enabling new kinds of services interactions between suppliers and buyers [5]. However, how can one ensure that suppliers and buyers have the same understanding regarding the issues that are subject of the negotiation?

Ontology is required to ensure that suppliers and customers (represented by software agents in the context of our work) are negotiating about the very same good/product/service.

3. THE ARCHITECTURE OF E-COMMERCE SYSTEM

This system adopts Java language to realize the prototype for it has transplant characteristic, which is easy to implement the mobile agent. In this architecture, there are two agent module introduced: agent server, agent adapter. This environment is installed in an agent server and has the power of creating the agent, copying the agent, migrating the agent and executing the agent.

It consists of subsystems at the buyer sites; the supplier sites, the agent adapter sites, and the agent sever sites, a bank. The buyer, supplier, agent adapter, agent sever, and mobile agents in each site interoperate in this system. We use mobile agents as mediators to automate a variety of tasks including buying and selling of products over the Internet.

The architecture of the system is as depicted in Fig.1.:

The first important part is Agent Adapter. It is a middleware used for initializing and managing agent as well as a broker server, it provide environment for the mobile agent to be survived and operated, it can be realized with

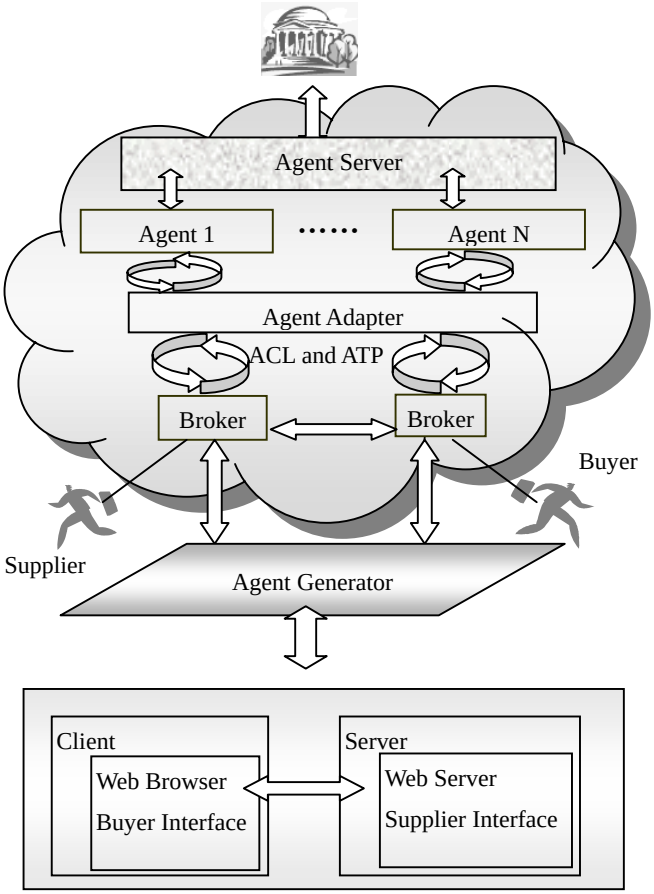


Fig.1. The architecture of E-Commerce System

Java. Users request for data from database with help of the adapter; the database replies data to agent's adapter. According to the application program from the demand of users, the agent adapter can establish a lot of agents for different tasks or the same task at the same time, and can run the task side by side. Under the drive of the task, it could realize the migration among every host computer of agent. When the host computer that operates agent meets an unexpected accident, it could store the status information of agent, and resume operation of agent when the host computer system resumes operating. Or because agent founder has cancelled the execution of this task after the task was finished, the agent adapter could destroy the agent. The agent adapter still offers naming service. The agent adapter carries on the identity and validation with agent server of the destination to exchange the authentication with the public key, so that the opposing side carries on correct decryption. The agent adapter carries out a series of packing, compressing, encrypting to mobile agent, and then implements the migration of agent. Agent server receives the package that migrated from the remote host and carries on various kinds of inverse transformation, decompression, and decryption. Finally, it starts up migrating function module of agent according to the task from agent adapter.

The second important is the Agent Server. It keeps the original data with local database, states form, storehouse form structure, and data range of the local data as such. Each agent server is registered to the agent adapter, and the original data of each agent server be known by the agent adapter through registering, thus it will allocate task in advance. It will offer the service facility for agent such as operating manager, the communication manager, security

manager, migration manager of mobile agent; it supports flow between every node of agent, and manages the operation working on the node of agent, and offer support communicating between every agent, offer identity authentication to each agent. Agent server generally adopts agent transfer protocol (ATP) and realizes agent migration among the host computers, and distributes the executing environment and service interface for agent; adopt agent communication language (ACL) to communicate with agent mutually.

The third important part is the ontology product database. The Customer Agent and the Supplier Agent have the same objective: they want to trade products in the same application domain, but both use their own private ontology. The ontology is used to design the product in the DAML+OIL language. Base on the shared ontology, a common vocabulary can be established to allow different type of agents in negotiation share in order to discuss the issues that can arise in the negotiation.

The form with initial e-commerce is to transmit information through Internet and E-mail, and then release vivid information of corporation through Web, then move to the interacting way and exchange information in order to realize online trade finally, it make the whole trade activity electrification. Mobile agent technology makes code, the data, and status information encapsulating one agent, and then migrates one host to another and guarantees its integrality of state attribute. At the same time, it ensures the independence of platform and security. So we choose Java to realize electronic payment system. Java allows the procedure not to be revised to run in different platform, and offer the programming interface oriented object. Remote Method Invocation (RMI) realizes distributing computing of the system. It can produce distributed Java application program, thus make Java object method on different host being transferred by other Java virtual machines. Once a Java program has obtained the index of remote object, it can inquire about the remote object method in servers through the naming service that is offered from RMI, or invoke remote object through the method, which regarding the received index as the parameter or the returning value. The customer can invoke the remote object in the server, and this server itself can be the customer of a remote object. RMI gathers and disperses the parameter through using the continuous object, and such doing does not reduce the type of the data, in this way; it can support object -oriented polymorphism. Every agent in the system has its one's own remote operation interfaces; each agent's needs to register on RMI server that this system operates. While the adapter carrying on the method to invoke, it could transmit the parameter directly to each agent, and obtain the result from agent directly.

An agent-based architecture for electronic commerce allows the creation of a virtual marketplace in which a number of autonomous or semi-autonomous agents trade services and goods. This approach offers a number of new opportunities to reduce or even eliminate the disadvantages mentioned above. Agents are able to examine a large number of products before making a decision to buy or sell [6]. This not only eliminates the need for manually collected information about products but also allows negotiating an optimal deal with the various sellers of a good.

3.1 THE BUYER AGENT

Customer's subsystem: Host computer of the customer (or the host computer of the merchant) dispatch broker agent (BA) to the host computer of server. The server's host computer produces the certificate and public key and the private key to the customer (or the merchant). At first customer's host computer dispatches Buyer Agent to the host computer of server, and obtains the key and address and return back. Then customer's host send buyer agent to inquire the key and address, which obtained from the merchant. After compared with various merchants, the buyer agent selects some merchant to negotiate. Agent chooses one merchant to deal finally. The framework is as following (Fig. 2.):

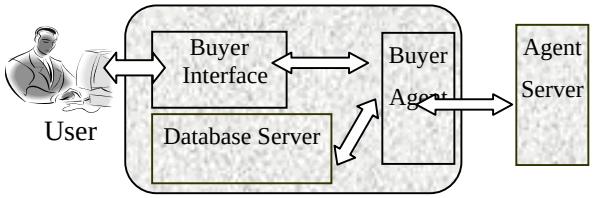


Fig. 2. The framework of the customer's subsystem

When a user creates a buying agent, a correct description of the item that he wants to buy must be given. Basically, the agent goes to the marketplace, contacting interested parties (selling agents) and negotiating with them the best possible deal taking into account different user's preferences. The user needs to specify a set of attributes when he creates the buyer agent like: Description of the item; Desired price; Highest acceptable price; Desired date to buy the item; Terms of guarantee; Satisfaction with certain providers; Delivery times acceptable (Physical goods); Authorization to do the transaction / need user approval; Other parameters.
The realizing flows as follows:
(1) Through web (Buyer Interface) user sends out visiting information to the buyer agent.
(2) Customer agent asks local query agent to finish the inquiry service of the database server in order to find item to be purchased.
(3) The buyer agent specifies the criteria for the acquisition of the product, and dispatches the mobile agent to the potential suppliers. The mobile agent visits each supplier site, searches the product catalogs according to the buyer's criteria, and returns to the buyer site with the best deal it finds and adds it to the database for the future using.
(4) Customer's host computer dispatches mobile agent to do shopping. According to address and key of the host computer of the merchant and carry encrypted Agent ID in shopping at the same time.
(5) Customer's host computer compares the information sent from every merchant, chooses the suitable trade company to consult, until both sides being satisfied.
(6) Customer Agent must undergo authentication through authentication first while doing a deal, the customer can get the address of the relevant merchant and public key information, it is mutual with agent of the merchant at the same time, send one's own public key and Agent ID to the merchant's agent. While waiting for the customer to send agent to do shopping, through decrypting and proving digital signature, they can do a deal after being unanimous.

3.2 THE SUPPLIER AGENT

Supplier's subsystem: It is an authentication center offering services, such as registration, inquiry, authentication of the customer and trade company that formed by electronic shop and customer, It can share resource and password tactics in common use of the market. Do a deal after registering to the host computer server. After a mobile agent to visit in a shop,

it should be from host server to get the user's key that the host computer obtains first, it could just be mutual with it after authentication. Systematic structure frame picture is depicted in Fig. 3.:

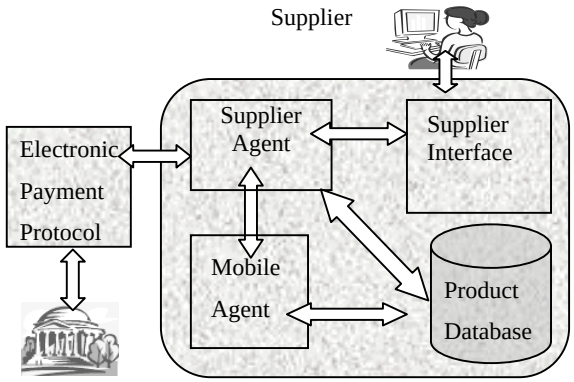


Fig. 3. The framework of the merchant's subsystem

In contrast there are the buying agents, which job is to buy goods on behalf of users. When a user creates a selling agent, he gives it a correct description of the item to sell. The user needs to specify a set of attributes when creates the seller agent like: Description of the item; Desired price; Lowest acceptable price; Stock available;- Desired date to sell the item; Delivery times (Physical goods);Terms of guarantee; Authorization to do the transaction / need user approval; Other relevant parameters.

The realizing flows as follows:

1) Merchant's Agent inquires about the local database in order to find the goods that the customer is interested in, if there is stock, it wills feedback information and give customer agent to transact immediately. If there is no stock, it will shift to the stock management system and file an application. There is source of goods that triggers merchant's agent to carry out and operate correspondingly in time.

2) The trade company, after verifying the certificates of the customer and carrying on corresponding digital signature information, obtained the decrypted agent, and compared agent which customer's server host computer send before, if they are the same, receive the service, otherwise refuse the service.

3) After the purchase intends of the buyer and seller confirmed, customer agent and merchant do a deal through the electronic payment arrangement, in the course of paying, should pay attention to guarantee the security of information of the agent [7].

Mobile agent system should have characteristic of being independent of any platform, so this model selects Java as developing languages in development and select Aglet [8] Workbench of IBM as development environment.

3.3 ONTOLOGY PRODUCT DATABASE

We provide a description method to establish ontology of the goods. It will often reflect in e-commerce trade that the same product name has different semantic description to different participants. We use ontology theory to define product name and its connotation in order to solve the semantic trade problem.

Step1 we should establish the purpose and range of the ontology product. In e-commerce we offer a unified ontology product system which solving differently semantic that the enterprise goods describe in order to fulfill the application systems of different enterprises to share the information of the goods.

Step 2: The ontology is obtained. We establish the key, important goods term in the goods field (such as the price, company), it last these definition of name by clear characters and between relations, describe with the file first.

Step 3: Formalization is defined, we establish the basic term (such as kind, instance and relation), these will be used in the explanation of the ontology and define, it will be selected to define the description language used of the ontology, we use the selected ontology language to describe goods finally.

Step 4: If we have chance to integrate with some existing ontology probably.

Step 5: Assess the ontology: the defined ontology assessing from a technological angle and assessment on utility in using defined before using.

Step 6: Set up the file of the ontology of the goods.

4. CONCLUSION AND DIRECTION FOR FURTURE WORK

Today there are several Web sites where people can look for a specific good or service; we believe that by using agent technology, ontology, web application tools and simulation techniques we can solve many of the problems presented [9]. We had presented a model for a virtual Marketplace where users create autonomous agents to buy and to sell goods or services and represent the interests of their owners. The intelligent agents automate the time consuming process of looking for buyers or sellers and can negotiate in order to obtain the best deal. Our model has many providers for a product or service and many clients that want purchase the product or service. Any transaction has specific parameters and is based in items like price, available stock, delivery policies, and terms of guarantee, satisfaction with certain providers, etc.

The proposed model in this paper provides a new solution to the development of intelligent e-commerce applications. The proposed integrated approach has shown its potential in the future electronic market. In our work, various information technologies are integrated to automate the e-commerce. These technologies include intelligent agent, ontology technology and web technologies. Our future work will be focused on developing some security mechanisms to provide security services for the system [10]. In order to realize the interoperating and the cooperation between every part of implementation system, the work needing to be perfect further has included protect mobile agent running in hostile environment and cooperate with other agent of hostility and so on.

We also want to explore some implementation details such as the communication language between the agents and the system, and the ontology problem. Even if a universal communication language were to be adopted, the participating agents would still only be using a common syntax.

5. REFERENCES

[1] Foundation for Intelligent Physical Agents, http://www.fipa.org.

[2] R. Guttman, A. Moukas et al, "Agent-mediated electronic commerce: A survey," Knowledge Engineering Review, 13(2) (June 1998), pp.147–159.

[3] A. Chavez, P. Maes, "Kasbah: An agent marketpla_

ces for buying and selling goods," Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK (April 1996), pp. 75–90.

[4] M. Tsvetovatyy, B. Mobasher, et al, "MAGMA: An agent-based virtual market for electronic commerce," Applied Artificial Intelligence, 11(6) (September 1997), pp. 501–523.

[5] D. Fensel, "Ontologies and Electronic Commerce", IEEE Intelligent Systems, January /February 2001, pp 8.

[6] R.H.Guttman, P.Maes, "Agent-Mediated Integrative Negotiation for Retail Electronic Commerece," Lecture Note in Artificial Intelligence 1571, Agent Mediated Electronic Commerce, 1998, pp. 70-90.

[7] Amitabha Das, G.X.Yao, "A Secure Payment Protocol Using Mobile Agents in an Untrusted Host Environment," ISEC 2001, LNCS 2040, 20018, pp. 33-41.

[8] Aglets, http://www.trl.ibm.co.jp/aglets/, 2004.

[9] T.O. Lee, Y.L. Yip, et al, "An Agent-Based Micropayment System for E-Commerce", E-Commerce Agents: Marketplace Solutions, Security Issues, and Supply and Demand, LNAI 2033, Vol: 2033 / 2001, pp. 247-263.

[10] D. M. Chess. "Security issues in mobile code". G.Vigna (Ed), Mobile agent and security, Lecture notes in Computer Science 1419, springer, Berlin, 1998.

Design of Distributed Wireless Sensor Networks in Campus

Shijue Zheng, Ying Su, Li Gao, Shiqian Wu
Department of Computer Science, HuaZhong Normal University
WuHan, 430079, China
Na Zhang, Jun Liang
Department of Computer Science, Guang Xi Normal University
Guilin, 541004, China
E-mail: suying929@163.com

ABSTRACT

The availability of micro-sensors and low-power wireless communications enable the deployment of distributed sensor networks for a wide range of applications. Considering the great potential of distributed wireless sensor networks in civil domains, this paper gives an idea about the application of wireless sensor networks in campus and analyzes the benefits that the sensor networks bring. Taking the Central China Normal University for example, we consider the specialty of campus and design the model of the campus sensor network in detail, which includes indoor and outdoor placement of sensor nodes. In addition, based on the placement an improved OPTIMUM-COV algorithm that applies in campus in the design was introduced. The aim is to achieve anticipant request of coverage by laying as less sensors as possible. At last it gives the simulation results and statistical data. The results show our design is feasible.

Keywords: wireless sensor network, placement, coverage, outdoor placement, OPTIMUM-COV algorithm.

1. INTRODUCTION

Recent advances in digital electronics, embedded systems and wireless communications are leading the way to a new class of distributed wireless sensor networks. The wireless sensor networks (WSNs)[1，2] combine the functions of sensing, data collection and storage, computation and processing, communication through a wireless medium, and/or actuating. WSNs have been considered and envisioned in a wide spectrum of applications in various military and civil domains. WSNs exhibit revolutionizing approaches to providing reliable, time critical, and constant environment sensing, event detection and reporting, target localization and tracking. Due to their ease of deployment, low cost, flexibility, and ability to self-organize, WSNs can be deployed in almost any environment, especially those where conventional wired sensor systems are impossible, unavailable or inaccessible, such as in inhospitable terrains, dangerous battlefields, outer space or deep oceans. Therefore, the existing and potential applications of WSNs span a wide spectrum in various domains, such as control, communications, computing, intelligence, surveillance, reconnaissance and targeting for military purposes; environmental detection and monitoring; disaster prevention and relief; medical care; home automation; scientific exploration; interactive surrounding, etc.

Now with the development of technique, the cost of sensor node is becoming lower and lower with its strengthening function, which makes the distributed wireless sensor networks that cost too much before can be used in more civil fields and facilitate our lives. Such as street lamp illumination, traffic control and intelligent home[3].
Considering the large areas, many buildings and complex facilities in campus, it will be very convenient for the college logistics to unification management and environmental monitoring by collecting the data of the noises, temperature, humidity, the light of street lamp and the use state of classrooms. On the basis of the characteristic of the sensor，this paper advances a design scheme, in order to utilize the sensor networks in college.

The remainder of the paper is organized as follows: In Section 2, we provide an investigation of wireless sensor networks. In Section 3, we establish the model of the sensor network in CCNU, and analyze it summarily. We present the placement of sensors and introduce an improved OPTIMUM-COV algorithm. In Section 4, the simulation work is realized on MATLAB. Finally we give the statistical data of sensors both indoor and outdoor placement. We conclude our paper in Section 5.

2. RELATED WORKS

The wireless sensor networks are composed of numerous sensors, which relate mutually, process and transmit information through the wireless communications. WSNs synthesized sensing, embedded computation, distributed information processing and the communication, which ensure a wide range of functions such as real-time monitor, sensate and gathering each kind of environment or the monitor object information in the region of the distributed network, then processing the information and transmitted to the user who needs.

2.1 Architecture of sensor network

The architecture of wireless sensor network is shown in Figure.1. The sensor nodes are usually scattered in a sensor field. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multi-hop infrastructureless architecture through the sink as shown in Figure. 1. The sink may communicate with the task manager node via Internet or Satellite.

* This work was supported by the National Natural Science Foundation of P.R.China under Grant No.60573118.

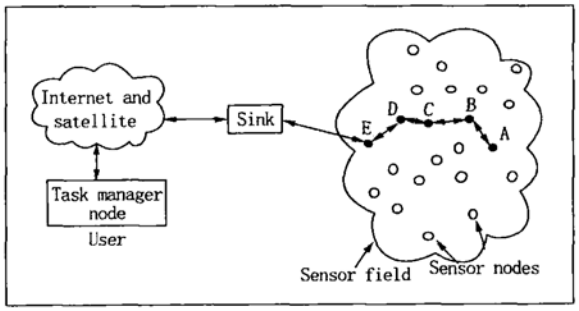


Fig. 1. Architecture of sensor network

2.2 Components in node of sensor

The wireless sensor network is made up with the multitudinous small and cheap sensor nodes，which have the functions of wireless communication, sensing and data processing. A sensor node is made up of four basic components as shown in Figure.2: a sensing unit, a processing unit, a transceiver unit and a power unit. They may also have application dependent additional components such as a location finding system, a power generator and a mobilizer.

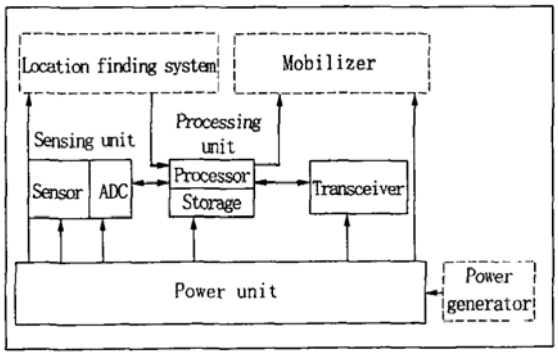


Fig. 2. Components in node of sensor network

2.3 The sensor networks protocol stack

The protocol stack used by the sink and all sensor nodes is given in Figure. 3. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane.

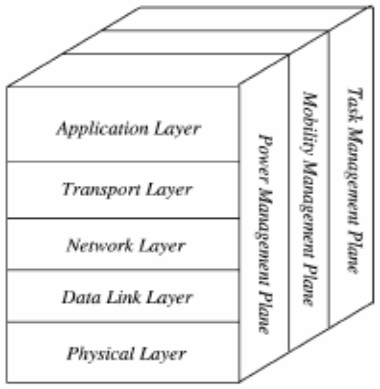


Fig. 3. Protocol stack of sensor network

3. DESIGN OF DISTRIBUTED SENSOR NETWORKS

This paper takes the Central China Normal University (CCNU) for example. CCNU has a campus of over 2,000 mu with a construction area of more than 600,000 square meters on Guizishan. At present the university has a student body of over 20,000, and teachers and faculty of about 4,000. If WSN is used in campus, it will be of great use for the three aspects as follows.

1) Conservation. There are a great deal of expenses of water and electricity everyday in the campus, and we can monitor the use of inside electricity, lighting of street lamp and the use of water, in order to save the resources

2) Security. There are many people in the campus with its big areas, so the security is an important issue, and with the WSN, we can take the circumstance in the campus in control betimes, and find out the factor that is not safe in order to advance the safe factor in the campus in time.

3) Management. We can get a great deal of information of each aspect through the WSNs. So it can make the management of logistics and security more convenient and efficient.

4) Others. It also brings convenience in other aspects. Through getting information by WSNs, we can have more effective arrangements of teaching and more timely renewal of equipment.

What's more, one of the most important constraints on sensor nodes is the low power consumption requirement in many applications. But in campus we can solve this problem by changing batteries in time or using the electric power supply. It also strengthens the feasibility of wireless sensor networks in campus.

3.1 The model of sensor network

The entire network is made up of some wireless sensor nodes and base stations, and it may consist of many different types of sensors such as seismic, low sampling rate magnetic, thermal, visual, infrared, and acoustic, which are able to monitor a wide variety of ambient conditions that include temperature, humidity, vehicular movement, lightning condition, pressure, noise levels, the presence or absence of certain kinds of objects, and the current characteristics such as speed, direction, and size of an object.

The sensor nodes have processing and wireless communication capabilities, which enable they to gather information about the environment and to generate and deliver report messages to the remote base station. The base station aggregates and analyzes the report messages received and decides whether there is an unusual or concerned event occurrence in the monitored area. Base station can be in a certain building and a mobile platform will as well as do. Base station consists of a computer and a wireless interface module.

Then the professional staff gets all data in computer center via the campus network. Map of CCNU campus and the topology of wireless sensor network in CCNU campus are shown following:

Fig. 4. Map of CCNU campus

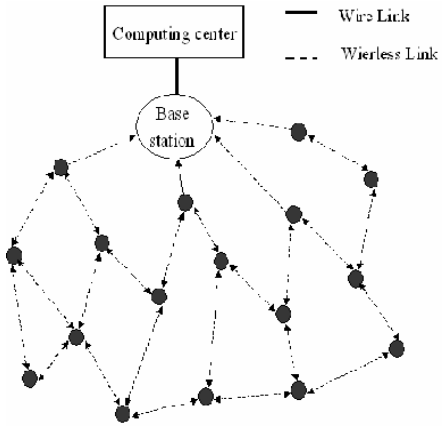


Fig. 5. Topology of WSN in CCNU campus

3.2 The placement of sensor nodes

Since it is the network in campus, we can place the sensors manually, and need not to think about whether the energy sources are exhausted because the batteries can be changed conveniently and in time. We just need to avoid blind areas of supervision and communication to satisfy the request of overcast areas. There are mainly two different aspects, one is the indoor placement[4] of sensors，and another is the outdoor one.

We can place sensors indoor just on the basis of our need of monitor, such as in the halls, the washing rooms, the area of dormitory ,the classrooms, the library and so on, without considering other factors.

While when it comes to the outside ground, we adopt the OPTIMUM-COV algorithm, which is an ameliorative one based on the MAX_AVG_COV algorithm and the MAX_MIN_COV algorithm mentioned in literature[5].

The aim of the OPTIMUM-COV algorithm is to achieve anticipant request of coverage[6] by laying as less sensors as possible. The basic thought is to place the sensors on the points where the integral effective coverage performance can be achieved, and eliminate local redundancy in order to upgrade integral performance. First the region where sensors will be laid is divided into grid. Then we suppose the probability that the sensors detect the objects varies exponentially with the distance between the goal and the sensor, and make a liberal allowance for influence of the obstacles.

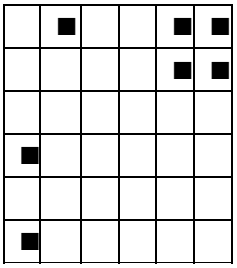


Fig.6 Grid of sensor region(■---obstacle)

In the algorithm, grid size N, granularity d as well as number k and position of obstacles are given. We suppose the probability that the sensor detect the object varies exponentially with the distance between the goal and the sensor, namely P_{ij}. P_{ij} denotes the probability that the sensor on point i discovers the object on point j. If there are obstacles on the line ij, P_{ij} is 0. When *i* equals *j* P_{ij} is 1. In other situation P_{ij} varies exponentially.

$$p_{ij}=\begin{cases}0 & obstacle\ on\ ij\\ \exp(-\lambda\times d) & others\\ 1 & i=j\end{cases} \quad (1)$$

α denotes the attenuation factor of the sensor nodes . Given the limit of sensors number N_{limit} and the undetected probability limit of crunodes $M_{min,}$ we can compute the matrix M ,

$n=N\times N$, $m_{ij}=1-p_{ij}$

$$M=\begin{bmatrix} m_{11} & m_{12} & \dots & \dots & m_{1n}\\ m_{21} & m_{22} & \dots & \dots & m_{2n}\\ \dots & \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots & \dots\\ m_{n1} & m_{n2} & \dots & \dots & m_{nn}\end{bmatrix} \quad (2)$$

And assign initial values to the matrix M*, M * = ones (N ,1). In it M_i denotes the undetected probability of the point i.

$M^*=(M_1\ M_2\ M_3\ \dots\ \dots\ M_n)$ (3)

If put a sensor node on point j,

$$E_j=\sum_{i=1}^{n} Max(M_i\times M_{ji}, M_{\min}) \quad (4)$$

The less the value of E_j is, the more improve the network's performance do. When the placement of sensor nodes are completed,

$$E=nM_{\min} \quad (5)$$

/* Call OPTIMUM-COV Algorithm, Complete the placement of sensors */

{ assign initial values to the parameters, such as n, M and M;*

*while(E!=n*M$_{min}$) {*

for(j=1;j<=n;j++) {

if (L(j)!=1) // not place a sensor node on j

{

compute the new values of M if place sensor node on j;*

construct a new matrix M(j);*

$E_j=\sum_{i=1}^{n} Max(M(j)^*\times M_{ji}, M_{\min})$;

}

```
    if ( E_j is the minimum ) {
        L( j )=1;   //place a sensor node on j
        Num=Num+1;
        M*=M( j )*;
        E=E_j;
      }
   }
  }
}
```

By means of algorithm iterating, we can obtain a series of mesh points. We place sensors on the points, and then obtain more effective coverage.

4. SIMULATION RESULTS

In this section, we present the performance analysis of OPTIMUM-COV algorithm by using simulation programs written in C. According to the design thought above, we take the CCNU campus for example and give the detailed situation about the placement of sensor nodes in CCNU campus.

First we divide the campus in a 32×32 grid in which the distance of every two points is d=35m by field measurement. And it is calculated that there are 58 obstacles in the grid. Then under the above given conditions, the outdoor placement called OPTIMUM-COV algorithm completes the placement of sensor nodes. α =0.1 and α =0.2 denote the different types of the sensors which have different parameters respectively. The simulation work is realized on MATLAB. The MATLAB simulation results are shown in Figure 7.

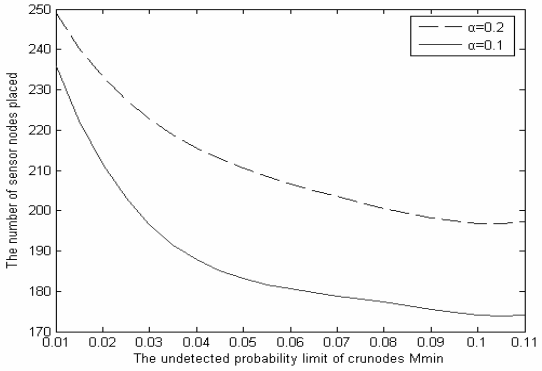


Fig. 7. Results of outdoor placement

Figure 6 presents the situation that the number of the sensor nodes increase when the undetected probability limit of crunodes changes. Meanwhile the value of α is bigger, the number of sensors needed is larger. In the actual work, we choose the sensors which α is 0.1.

In addition, on the basis of practical application, sensor nodes are placed in each construction of CCNU campus on the basis of our need of monitor. Finally the statistical data of sensors both indoor and outdoor placement are shown in the following table.

Table 1. The number of sensors

Outdoor	M_{min} =0.01	237
Indoor	Office Building	20
	Classroom Building	90
	Residence	212
	Cafeteria	35
	Gymnasium	4
	Library	9
Total		607

5. CONCLUSIONS & FUTURE WORKS

The distributed wireless sensor networks was integrated with many advanced techniques in order to supply the way of getting information, processing information for people. Meanwhile it also advances a great deal of new challenges. Due to their ease of deployment, low cost, flexibility, and ability to self-organize, WSNs can be deployed in our daily life. By introducing the wireless sensor networks and its applications in campus, we give a feasible design of distributed wireless sensor network in detail. The experimental results show that the outdoor placement of sensor nodes achieves a good coverage performance. But there is no similar application that utilizes the technique of distributed wireless sensor networks in campus at present. In addition, routing protocol is needed to improve the performance of the wireless sensor networks. Routing protocol is not considered in this paper. We are planning to cover this topic in future work.

6. REFERENCES

[1] I.F. Akyildiz, W. Su et al, “Wireless sensor networks: a survey”, Computer Networks, Vol.38, 2002, 393-422.

[2] D. Ganesan, A. Cerpa, et al, “Networking issues in wireless sensor networks”, J. Parallel Distrib. Comput., Vol.64, 2004, pp.799-814.

[3] J.G.Zheng, C.D.Wu, et al, “The Application of Wireless Sensor Network in Intelligent Home”, Intelligent Building, Vol.8, 2005, 24-26.

[4] Jian Tang, Bin Hao et al, “Relay node placement in large scale wireless sensor networks”, Computer Communications, Vol.29, 2006, 490-501.

[5] Santpal Singh Dhillon, Krishnendu Chakrabarty, “Sensor placement for effective coverage and surveillance in distributed sensor networks”, Wireless Communications and Networking, 2003, WCNC 2003. 2003 IEEE [C], 3(3), 2003, 1609-1614.

[6] Y.L.Wang, S.Q.An, “Research on the Coverage of Wireless Sensor Network”, Chinese Journal of Sensors Actuators,18(2), 2005, 307-311.

A New Micropayment Scheme in Mobile Data Network *

Yining Liu
Faculty of Mathematics and Computer Sciences, Hubei University
Wuhan, Hubei Provience 430062, China
Email: lyn7311@sina.com
Jinbing Tian
Faculty of Mathematics and Computer Sciences, Hubei University
Wuhan, Hubei Provience 430062, China
Email: hubu_tj@yahoo.com.cn
Hongwei Li
School of Mathematics and Physics, China University of Geosciences,
Wuhan, Hubei Province 430074, China
Email: hwli@cug.edu.cn

ABSTRACT

Micropayment in mobile data network has been a hot topic in e-commerce field. The PayWord scheme of Rivest and Shamir utilizes linear hash chain to ease the burden of authentication. The Quan Son Nguyen scheme generalized linear hash chain to multi-dimensional hash chain in WCC2005. In this paper, we analyze Quan's scheme is infeasible, and give a new micropayment scheme in mobile data network , such that computational overheads on the mobile device which usually has limited computational capability and storage is minimized.

Keywords: hash function, hash chain, node, micropayment.

1. INTRODUCTION

Over recent years, there has been brought to proliferation of the Internet as the network technologies develop and accordingly the demands for use of the Internet increase. The Electronic Commerce (E-Commerce) is derived from the desire that people want to purchase information or goods by means of the Internet technologies. Micropayments are electronic payments of low value and they are now playing a major role in the expansion of electronic commerce. At present, a large portion of electronic commerce occurring in the mobile data network belong to the category of micropayments, such as ringing tone download, news subscription, etc. Micropayment system's designs are usually quite different from the existing "macro-payment" systems, since micropayment in mobile data network must be very simple, secure, and efficient, with a very low overhead cost per transaction.

The simplest form of payment scheme is an electronic check. A user pays a merchant for a transaction by digitally signing a piece of data that identifies the transaction (the user, the merchant, the goods, the amount to be paid, the time, etc) and possibly other data (such as account and credit information). The merchant deposits an electronic check by sending it to the bank. Having verified that the check is genuine and that has been deposited for the first time, the bank credits the merchant with the proper amount and charges the buyer with the same amount.

In principle, micropayments could be implemented by electronic checks. Merchant and user are engaged in the transaction, and the computation time they may devote to digital signing is not a real problem. The real problem lies with the bank's processing cost of any form of payment. It sometimes seems that the cost of doing any transaction with the bank will be many times lager that the value of the micropayment itself. For example, processing a credit card transaction costs about 50cents today, while a typical micropayment may be worth 5 cent. So in micropayment schemes, our main goal is to minimize the number of public key operation required per payment, using hash operation instead whenever possible. As a rough estimate, hash functions are about 100 times faster than RSA signature verification, and about 10000 times faster than RSA signature generation [1].

The PayWord scheme of Rivest and Shamir utilizes linear hash chain to ease the burden of authentication. The scheme of Quan Son Nguyen generalizes linear hash chain to n-dimensional hash chain (we call it multi-dimensional hash chain, short for MDHC)[2].

In this paper, we analyze the Quan's Scheme is impractical at first, and modify it to adopt the need of PayWord. This paper is organized as follows. In Section 2, we introduce the PayWord and Quan's n-dimensional hash chain scheme. In Section 3, we analyze why Quan's scheme is impractical. In Section 4, we modify and generalize Quan's multi-dimensional hash chain. Finally, Section 5 concludes the paper.

2. PayWord AND Quan's MDHC SCHEME

2.1 PayWord

The PayWord micropayment scheme is based on a chain of hash values, called *paywords*. Each *payword* represents a unit of small value, for instance a cent. Before transaction between a user and a vendor, the following steps need to be done.

1) The user selects a random value x_n and generates a hash chain as follows:

$$x_0 \leftarrow x_1 \leftarrow x_i \ldots \leftarrow x_{n-1} \leftarrow x_n$$

where $x_i = h(x_{i+1})$ for $i = 0,1,\ldots,n-1$, and $h(x)$ is a one-way hash function. x_n is called the "seed" and x_0 is

* This research is supported by the Research Foundation for Outstanding Young Teachers, China University of Geosciences (Wuhan) No.CUGQNL0607.

called the "root" of the hash chain. User commits the signature of x_0 and other information to the vendor. *M* is a data signed by the user's private key. It consists of x_0, vendor's identity (*V*), C_U, and other information (*I*). For example, $M = Sign_U(x_0, C_U, V, I)$. The vendor can authenticate the identity of the user because no other people know the private key of signature.

2) When the user wants to purchase some service or information goods, he can send (x_i, i) to Vendor as a *payword* coin. Vendor verifies whether x_{i-1} equals to $h(x_i)$ or not. Vendor accepts x_i as a valid payment only if the verification is successful. The vendor stores a valid x_i in place of x_{i-1}.

2.2 Quan's MDHC scheme

In Quan's sheme , Quan first generalize $x_0 = h^n(x_n)$, to $x_0 = h_1(h_2(...(h_n(x_n))...))$,then

$$X_0 = h_1^{n_1}(h_2^{n_2}(...(h_m^{n_m}(X_N)))) = \prod_{i=1}^{m} h_i^{n_i}(X_N).$$

In order to ensure the output is uniquely determined regardless of the application order, hash functions $h_1, h_2, ...h_m$ must be in pairs commutative, i.e. $h_i(h_j(x)) = h_j(h_i(x))$ for any x.

In Rivest's PayWord scheme, the storage-computational complexity is $O(N)$, and with the assumption of $h_i(h_j(x)) = h_j(h_i(x))$, the storage-computational complexity of Quan's scheme is $O(\log_2 N)$, which is more efficient . For the need of commutative, Quan selected RSA as hash functions.

3. ANALYSIS

Universal hash functions don't satisfy the property of commutative. Hash function is a deterministic function which maps a bit string of an arbitrary length to a hashed value which is a bit string of a fixed length. Let h denote a hash function whose fixed output length is denoted by $|h|$, some algorithm are proposed as hash function algorithm, e, g.MD5 or SHA. h Should have the following properties [3]:

3.1 Mixing-transformation

On any input x, the output-hashed value $h(x)$ should be computationally indistinguishable from a uniform binary string in the interval $[0, 2^{|h|}]$.

3.2 Collision resistance

It should be computationally infeasible to find two inputs x, y with $x \neq y$ such that $h(x) = h(y)$. For this assumption to be reasonable, it is necessary that the output space of h should be sufficiently large. The least value for $|h|$ is 128 while a typical value is 160.

3.3 Pre-image resistance

Given a hashed value h, it should be computationally infeasible to find an input string x such that $h = h(x)$. This assumption also requires the output space of h be sufficiently large.

Given input string x, the computation of $h(x)$ can be done in time bounded by a small-degree polynomial (ideally linear) in the size of x.

For the above property, it is obvious that $h_i(h_j(x)) = h_j(h_i(x))$ is computational infeasible for universal hash function, and the probability is very low, etc. $\Pr[h_i(h_j(x)) = h_j(h_i(x))] \leq \varepsilon$.

In Quan's scheme, Quan selected RSA as hash function, $h_i = x^{c_i} \bmod M$ and $h_j = x^{c_j} \bmod M$, it is true that $h_i(h_j(x)) = h_j(h_i(x)) = x^{c_i c_j} \bmod M$, but RSA is less efficient than universal hash. With it, micropayment system cannot burden the cost of public-key operation.

For above analysis, we conclude Quan's multi-dimensional hash chain is impractical.

4 MODIFICATION AND GENERALIZATION OF Quan's SCHEME

4.1 Modification of Quan's scheme

Assume that n denotes the total number of the pay-words in the chain, and $n = pq$. In this technique, two different hash functions are required. The nodes along axis X are generated using hash function h_1 and the nodes along axis Y are generated using hash function h_2. Each node represents a pay-word as a hash value. Moreover, the nodes $X'_0, ..., X'_{q-2}, X'_{q-1}$ represent the root of the chain which is used to verify the pay-word is true of false. The computing rules of each node can be concluding as follows:

1) Input the parameters

$X_n, p, q, i (i \in 1,2,...n)$, Where i represents the i th pay-word.

2) Compute $a = q - \lceil i / p \rceil$,

$$b = (p - (i \bmod p)) \bmod q, X_i = h_1^b h_2^a (X_{pq}))$$

3) Output X_i

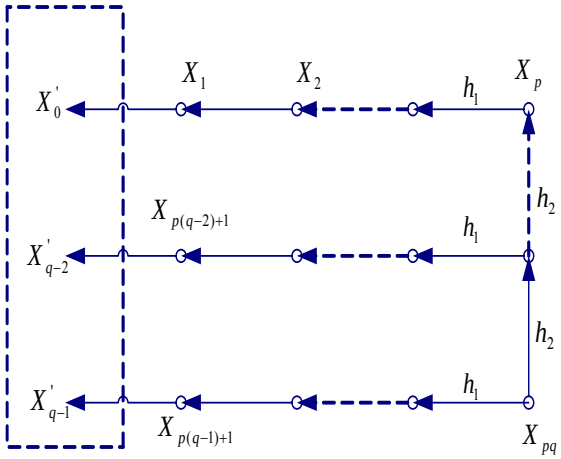


Fig. 1. The construction of the 2-dimensional hash chain

4.2 Generalization of Quan's scheme

In below scheme, we generalize 2-dimensional hash chain to m -dimensional hash chain. In the m -dimensional hash chain, each node represents a hash value. The numbers of nodes in the dimensions are, $d_1, d_2, ..., d_m$, and the nodes in the each dimension are derived by using the different hash functions $h_1, h_2, ..., h_m$ respectively. After the construction of the m -dimensional hash chain is defined (similar with 2-dimensional hash chain), the algorithm for computing node X_i in the m -dimensional hash chain can be generated as follows:

1) Input the seed $X_n (n = d_1 \times d_2 \times \cdots \times d_m)$ of the chain, and the index i of the hash value X_i

2) Let $a_0 = i, c_0 = X_n, d_{m+1} = 1$

For $j = 1$ to m , compute

$$a_j = a_{j-1} \bmod (d_j \times \cdots \times d_m)$$

If $a_j \neq 0$,

Compute

$$b_j = d_j - \left\lceil \frac{a_j}{d_{j+1} \times \cdots \times d_m \times d_{m+1}} \right\rceil$$

And $c_j = h_j^{b_j}(c_{j-1})$

If $a_j = 0$, break the for loop and the current node $X_i = c_{j-1}$

Until $j = m$, the current node $X_i = c_m$

3) the output is the X_i

5. CONCLUSIONS

1) For simplicity, in above 2-dimensioanl structure, let $p = q$, user needs $\sqrt{n} - 1$ hash function operations to generate a node on average. In m -dimensional structure, let $d_1 = d_2 = \cdots = d_m$, user needs $m \times (\frac{n^{1/m} - 1}{2})$ hash operations for computing each node averagely ,however user needs $\frac{n-1}{2}$ hash operation with Rivest's scheme.

2) Rivest's scheme requires a single hash function, while our scheme requires two different hash functions.

3) Rivest's scheme delivers one root to merchant; our scheme sends q roots to merchant for verification.

In one word, our scheme is more efficient than Rivest's PayWord and Quan's MDHC scheme, especially when n is large.

6. REFERENCES

[1] R.Rivest, A.Shamir, PayWord and MicroMint: two simple micropayment schemes. Proceeding of the 4th Security Protocols International Workshop, pp.69-87, Lecture Notes in Computer Science Vol. 1189. Springer -Verlag, Berlin,1996

[2] Q.S.Nguyen, Multi-Dimensional Hash Chains and Application to Micropayment Schemes.In International Workshop on Coding and Cryptography (WCC'2005), March 2005 in Bergen, Norway

[3] W.B.Mao, Modern Cryptography: Theory and Practice. Publishing House of Electronics Industry.2004,300-302

Load-Sensitive Traffic Balancing in Wireless Mesh Network *

Xuyang Ding, Mingyu Fan, Xiaojun Lu, Dayong Zhu, Jiahao Wang
School of Computer Science and Engineering, University of Electronic Science and Technology of China
ChengDu, SiChuan, China
Email: dingxuyang@hotmail.com

ABSTRACT

Although there is more than one access point (AP) in a wireless mesh network (WMN), the traffic load of each AP is quite different. An AP may be extremely busy during a particular period while the others are in very light load due to the inherent characteristic of routing algorithm of WMN. Traffic load concentrates on an AP may lead to network congestion, increase the packet loss rate and degrade the performance of network. It is very important to provide a mechanism to guide part of traffic load from busy area to light load APs and balance the traffic load of network. A novel traffic balancing mechanism is proposed to solve this problem. The mechanism discovers the heavy load nodes through a traffic load detection scheme, and broadcasts the load messages to every node. Then a node can choose communication paths according to the messages and avoid heavy load nodes or APs. The simulation results indicate that the traffic balancing mechanism can enhance the performance of WMN.

Keywords: Wireless Mesh Network, Traffic Balancing, and Traffic Load Detection

1. Introduction

Local area network system is widely implemented nowadays to provide hot spot coverage and the using of wireless mesh network (WMN) [1] further extends its applicability. WMN is an emerging disruptive networking architecture. In contrast to mobile ad hoc network (MANET), nodes of a multi-hop wireless mesh network are stationary and not constrained by energy.

Active research has been pursued for WMN to address issues of capacity, fairness, QoS, MAC and routing protocols, etc. Unlike MANET, traffic in WMN concentrates on the area around the APs and the throughput of each node decreases as O (1/n), where n is the total number of nodes in the network [2]. Of cause, there is more than one AP in WMN and a mobile node can choose a feasible path leading to another AP while it has been aware of the congestion. But we still need some protect mechanisms to take care of the coordination among the APs for that a node can change the path that may lead to different APs without worrying about the interruption of ongoing communications (similar to soft-handover in CDMA system).

In order to route packets from APs to end users, routing algorithm from MANET can be adopted for the similarity between WMN and MANET. However, due to the routing protocols proposed for MANET usually choose shortest path between APs and end users (such as DSR [3], AODV [4]), there is a serious problem that some APs and middle nodes are overloaded during particular periods while the rest APs and middle nodes have very low traffic load. These overloaded APs and middle nodes, or experience a long delay in the queue may drop some packets. So, if it is possible that some of traffic load going through heavy load paths could be routed to light load paths, the whole performance of the network can be improved.

In [5,6,7], some traffic balancing mechanisms are proposed to explore the wasted and unused bandwidth in MANET provides solutions to the previous problem by forcing some traffic load away the congested APs. However, the mechanism lacks effective management method to provide reference information for choosing new paths. Some researchers give weight to each hop and let the end users choose the path by looking at the sum of the weight. [8] Uses local queue size as weight and chooses the path with least sum of the weights. It is possible that nodes have an empty queue but the busy nodes nearby already block them. Even with a small sum of weight, the path may still go through a busy area of the network. [9,10,11] collect neighbors' information by listening to the periodical Hello message generated by the neighbor nodes. [9] Considers path load by the queue size of local and nearby nodes. [10] counts weight by the number of concurrent transmissions nearby [11] calculates the weight by adding the number of transmissions in local and nearby nodes. Among these proposals, the interference to a node may come from the area beyond its transmission range, thus the node can not know all information about bandwidth occupation around this area. The default path may still go through the busy area of the network.

To balance the traffic load of WMN and avoid end user connect to busy nodes or APs, this paper proposed a novel traffic balancing mechanism which can provide a more efficient way by detecting the load of routes and choose new paths, and guarantee the paths go through least number of busy nodes or area.

The remaining of this paper is organized as follows. Section 2 introduces the traffic balancing mechanism and shows how it detects the traffic load of network and solves the problem. Simulation results are shown in section 3 to indicate the advantage of traffic balancing mechanism in packet loss rate. Finally, section 4 concludes the paper.

2. Traffic Balancing Mechanism

Fig.1 gives a basic idea of the traffic balancing mechanism. Node A communicates with C. According to the DSR protocol, the selected path is likely to be the bold lines. It will pass through node B. But there are others packets also will pass through node B, such as node E and F communicates with D and G. The network will be congested when the node B's traffic load is high. Some packets are dropped because they exceed the lifetime or the buffer of node B is full.

Traffic balancing mechanism is a routing approach that tries to exploit the unused resources around lightly loads areas. For example, instead of selecting the solid line path

* The National Natural Science Foundation of China under Grant No.60373190 supported this research.

between node A and C in Fig. 1. , the path indicated by the dotted line is used for packets delivery for node A and C, and decreases the load of node B. This can either expand the network ability to support more communications under the same performance level or to improve the performance of the network. The mechanism is based on a reactive routing protocol and it can be implemented with any reactive routing protocol.

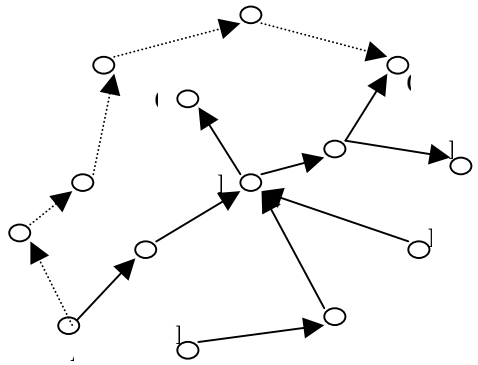


Fig. 1. Path Selection

For implementing the traffic balancing mechanism, it needs an effective solution to detect the traffic load situation of network and to give a reference for the new path selection.

2.1 Traffic Load Detection Scheme

Conventionally, WMN can be represented by an undigraph in the following manner. We assume that nodes of the network have the same transmission range. The topology of such a mesh network is modeled as a graph G= (V, E) where vertices (in V) represent individual mobiles nodes, and where an edge (in E) is placed between two vertices if the corresponding nodes are within transmission range of each other. For describing the traffic load detection scheme, we assume the network's topology likes Fig. 2.

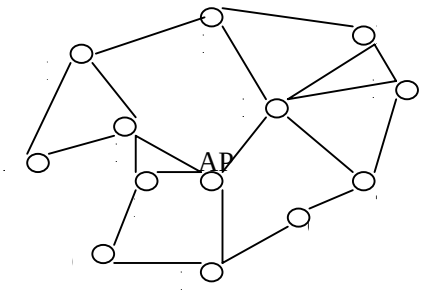


Fig. 2. Topology of Network

We define the routing weight is the least bandwidth of communicating requirement. For example, if an application needs the least bandwidth R (Mb) over a path, the weight of the path is R. Every node reports its routing paths and routing weights information to the nearest AP in a fixed time interval. For describing the scheme, we assume that there are six routes, H-I-AP-F (r1, weight is 1), A-D-AP-I-J (r2, weigh is 1.5), L-I-AP-E (r3, weigh is 1), K-I-AP-E-C (r4, weigh is 0.5), B-H-I-J-G (r5, weigh is 1) and D-AP-I (r6, weigh is 0.8). Especially, the route r1 and r3 are really communication paths, and the route r2; r4, r5 and r6 are the new applications' requests. According to below rules structure the network incidence matrix.

(i) Matrix $M_{n\times r}$ describes nodes and routes relations, where n represents the number of nodes and r is the count of routes. If route j passed node i, the entity $M_{n\times r}[i,j]$ is 1, otherwise is 0. So, we can get the $M_{n\times r}$ under previous assumption as (1).

$$M_{n\times r}=\begin{array}{c|cccccc} & r1 & r2 & r3 & r4 & r5 & r6 \\ \hline A & 0 & 1 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 1 & 0 \\ C & 0 & 0 & 0 & 1 & 0 & 0 \\ D & 0 & 1 & 0 & 0 & 0 & 1 \\ E & 0 & 0 & 1 & 1 & 0 & 0 \\ F & 1 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 1 & 0 \\ H & 1 & 0 & 0 & 0 & 1 & 0 \\ I & 1 & 1 & 1 & 1 & 1 & 1 \\ J & 0 & 1 & 0 & 0 & 1 & 0 \\ K & 0 & 0 & 0 & 1 & 0 & 0 \\ L & 0 & 0 & 1 & 0 & 0 & 0 \\ AP & 1 & 1 & 1 & 1 & 0 & 1 \end{array} \quad (1)$$

(ii) Matrix $M_{r\times w}$ represents the relationships of routes and weights of routes. Where r is the number of routes and w is the weights of the nodes. If route i used node j, then the entity $M_{r\times w}[i,j]$ is the weight of the route, otherwise is 0. we can get the $M_{r\times w}$ as (2).

$$M_{r\times w}=\begin{array}{c|cccccccccccc} & A & B & C & D & E & F & G & H & I & J & K & L & AP \\ \hline r1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ r2 & 1.5 & 0 & 0 & 1.5 & 0 & 0 & 0 & 0 & 1.5 & 1.5 & 0 & 0 & 1.5 \\ r3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ r4 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0.5 \\ r5 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ r6 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0.8 \end{array} \quad (2)$$

(iii) The product $M_{n\times r}$ and $M_{r\times w}$ gives a new matrix $C_{n\times n}$. It reflects the weight relationships of nodes by routes.

$$C_{n\times n}=\begin{array}{c|cccccccccccc} & A & B & C & D & E & F & G & H & I & J & K & L & AP \\ \hline A & 1.5 & 0 & 0 & 1.5 & 0 & 0 & 0 & 0 & 1.5 & 1.5 & 0 & 0 & 1.5 \\ B & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ C & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0.5 \\ D & 1.5 & 0 & 0 & 2.3 & 0 & 0 & 0 & 0 & 2.3 & 1.5 & 0 & 0 & 2.3 \\ E & 0 & 0 & 0.5 & 0 & 1.5 & 0 & 0 & 0 & 1.5 & 0 & 0.5 & 1 & 1.5 \\ F & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ G & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ H & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 1 \\ I & 1.5 & 1 & 0.5 & 2.3 & 1.5 & 1 & 1 & 2 & 5.8 & 2.5 & 0.5 & 1 & 4.8 \\ J & 1.5 & 1 & 0 & 1.5 & 0 & 0 & 1 & 1 & 2.5 & 2.5 & 0 & 0 & 1.5 \\ K & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0.5 \\ L & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ AP & 1.5 & 0 & 0.5 & 2.3 & 1.5 & 1 & 0 & 1 & 4.8 & 1.5 & 0.5 & 1 & 4.8 \end{array} \quad (3)$$

The diagonal elements of matrix $C_{n\times n}$ reflect the traffic load state of nodes. Assume the bandwidth of each node in the network is 2Mb/s. According to the previous assumption, it is obviously that the nodes D, J and I is overload. Besides, the others elements of $C_{n\times n}$ is useful in finding the heavy load or overload paths. For example, the value of $C_{n\times n}[I,D]$ is bigger than 2, so we can conclude that some packets through the path D-AP-I will be dropped because the high load of the path.

2.2 New Path Selection Scheme

Under the protection of a mechanism that taking care of the coordination among APs, nodes can choose a feasible path without worrying about the interrupt of the ongoing communications.

Through the traffic load detection scheme, an AP can obtain the traffic load situations of its neighbor nodes. Then, the AP broadcasts the heavy load and overload nodes' information to them. If a node's routing path includes the heavy load or overlod nodes, it tries to find another path to the destination without using these nodes. Similarly, when a node initiates a new communication, it should take care of the traffic load situation of nearby network according to the AP's broadcasts, and tries its best to avoid the heavy load or overload nodes. It is effectively to decrease the congestion of network.

In the new path selection scheme, it is necessary that a node make decision by taking into account the effect of its decision to the overall network. If a node chooses a new path in the same subnet of an AP, it only needs avoid this heavy load or overload nodes of the subnet. But when the AP is heavy load or overload, the node should choose another AP for communication. In this situation, the node needs to know the traffic situation of new subnet before it initiates new applications.

Simulation results proved that our traffic balancing mechanism gives an effective solution to solve the uneven traffic load problems.

2.3 Complexity Considerations

For computing the Time complexity of the traffic load detection scheme, we assume m = r = n. In fact, all nodes of network cannot communicate simultaneously because the wireless interference. So, m is always lager than r when we use the traffic load detection scheme to detect the traffic load of network. The time complexity of the scheme is as equation (4).

$$T(n) = O(n^3) \tag{4}$$

On the other hand, there is more than one AP in WMN. We can divide the network to several subnets according to the amount of APs. With the increasing of the amount of APs, the computational complexity of the traffic load detection scheme will decrease sharply as shown in Fig.3.

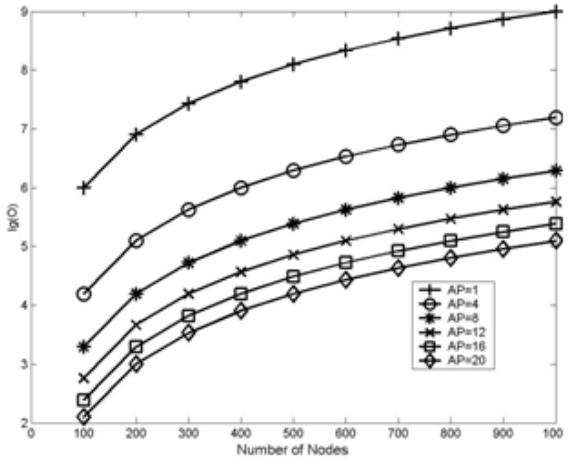


Fig. 3. Computational Complexity

3. Simulation Results

To evaluate the performance of the traffic balancing mechanism, the OPNET network simulator is used for the experiments. 50 nodes are randomly distributed in a 1000× 1000m^2 area with 2 APs locating at (350,350) and (650,650). CBR (Constant Bit Rate) is used as the data generating application and UDP (User Datagram Protocol) is used as the transport protocol. MAC layer adopts IEEE802.11b protocol and route protocol adopts DSR [3]. Every node's maximal traffic capability is 2 Mb/s and AP's capability is 24 Mb/s. Each of the sources generates data at the rate of 100Kb/s to a wired node and receives 100Kb/s CBR traffic from a wired node through one of two APs. The scenarios considered for the simulation can be classified with different node mobility, maximal node speed is 10m/s and 0m/s.

In addition, in the 802.11b standard, due to the collision avoidance mechanism provided by RTS/CTS exchanges, the effective capacity of the wireless link decreases when there is more than one node within the same collision domain. In simulation, we assume the radio interfering range is the same as the transmission range. Previous study by Li et al [12] has shown that the effective capacity of a chain topology becomes just 1/3 of the effective capacity of a single cell connection.

Considering the interference of wireless network, if the traffic load of a node is lower than 0.8 Mb/s, the level is normal, otherwise is heavy load or overload.

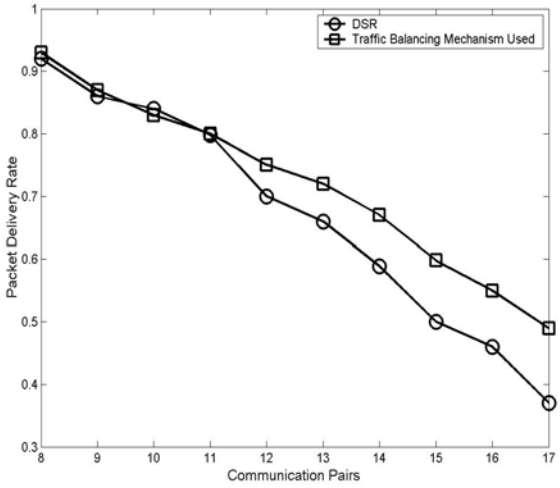


Fig. 4. Compare of Packet Loss Rate (10m/s)

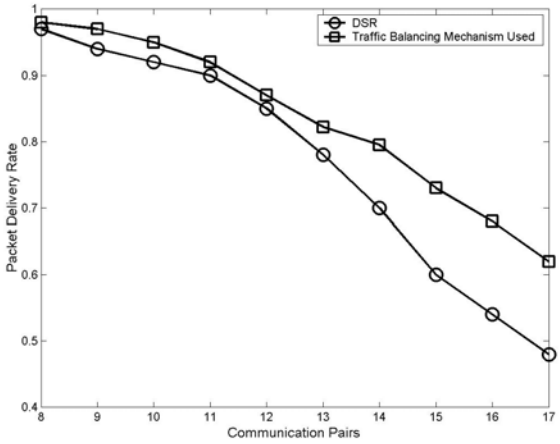


Fig. 5. Compare of Packet Loss Rate (0m/s)

Fig. 4. and Fig. 5. compare the performance in packet delivery rate between traffic balancing mechanism and DSR in different cases. The results indicate that the traffic balancing mechanism provides better performance. The improvement in static scenario is better than mobile scenario, because paths are more stable and the traffic balancing

mechanism can get more benefits from changing path. When only a few communication pairs (less than 12 in our simulation) exist in the network, few nodes are heavy loaded, so the improvement of traffic balancing mechanism is slight.

Table 1. shows results (16 Communication Pairs) of the simulations and indicates the APs traffic load situation. It is obviously that some of packets can be rerouted from busy AP to idle AP and the traffic load of network is more balanceable in our scheme.

Table 1. Traffic Load at APs

Class (Total 4048 Packets)	Mechanism	Number of Packets Transferred by AP1	Number of Packets Transferred by AP2	Packets Loss Rate
10m/s (16 Pairs)	DSR	1549	835	41.1%
	Traffic Balancing Mechanism Used	1518	1194	33%
0m/s (16 Pairs)	DSR	1844	993	30%
	Traffic Balancing Mechanism	1727	1471	20.5%

We also need to note that traffic balancing the throughput of WMNs also relates to the position of access points when the number of APs is more than one. Because less hops for each path leads to a better throughput in WMN. So, the best place for APs is the place where the overall average number of hops of all connections is smallest. Fig. 6. shows the relationship between the position and the network performance on packet delivery rate when the number of communication pairs keeps constant (14 communication pairs) and the network is static. There are 4 APs in the network and x-axis in Fig.6 is distance from each AP to the middle point of the network in horizon direction.

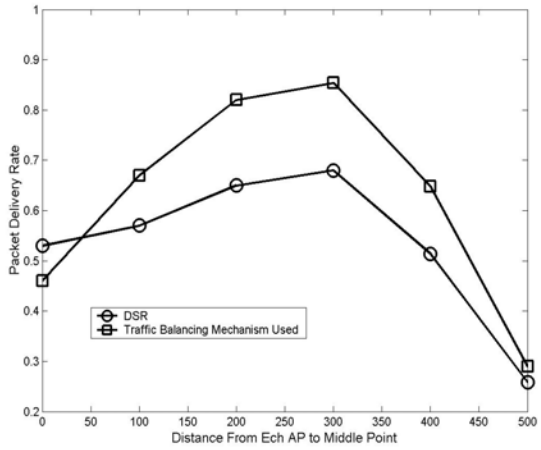


Fig. 6. Performance with APs in Different Places

Fig. 6. shows that the performance of both DSR and traffic balancing mechanism increases when APs locate from the range 0 to 200 and decreases when AP locate from 300 to 500. The reason is interference at the corner is higher and all traffic may go through same nodes before reaching APs. It is the same that the performance also degrades when all APs closely. Moreover, Fig.6 also shows that the performance of traffic balancing mechanism is better than DSR to solve the uneven traffic load problem when APs are at the appropriate place (the area close to the network's middle point).

4. Conclusion

Traffic load concentrates on an AP may lead to network congestion, increase packet loss rate and degrade performance of the network. The traffic balancing mechanism can effectively guide part of the traffic load from busy area to light load APs and balance the network's traffic load. Each node reports its routing paths and routing weights information to the nearest AP in a fixed time interval. The traffic load detection scheme discovers these heavy load nodes in the network and broadcasts the messages to every node. Then a node can choose communication paths according to the message and avoid the heavy load nodes. The simulation results indicate that the traffic balancing mechanism can enhance the performance of network while the APs are in the appropriate places.

REFERENCES

[1] I.F. Akyildiz, X.Wang, et al, "Wireless Mesh Networks: A Survey", Computer Networks Journal (Elsevier), March 2005,Vol.47, Pages 445-487.

[2] J.Jun, M.L.Sichitiu, "The Nominal Capacity of Wireless Mesh Networks", IEEE Wireless Communications, October 2003, pp.8-14.

[3] J.Broch, D.Johoson, et al, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", IETF Internet draft, Oct. 1999.

[4] C.E.Perkings, E.M.Royer, "Ad-hoc On Demand Distance Vector Routing", Proceeding of 2nd IEEE Workshop on Mobile Coputer Systems and Applications, Feb.1999, 90-100.

[5] X.Tao, T.Kunz, et al, "Implementing Traffic Balancing for Exploring System Capacity in Wireless Ad Hoc Networks", presented at WWRF12, November 2004.

[6] X.Tao, T.Kunz, et al, "Adaptive Traffic Balancing in Wireless Ad Hoc Network", presented at WWRF13, March 2005.

[7] X.Tao, T.Kunz, et al, "Traffic Balancing in Wireless Mesh Network", International Conference on Wireless Networks, Communications and Mobile Computing, 2005.

[8] A.Zhou, H.Hassanein, "Load-Balanced Wireless Ad Hoc Routing", Proceeding of Canadian Conference on Electrical and Computer Engineering 2001, Vol.2, 1157-1161.

[9] K Wu, J.Harms, "Load-Sensitive Routing for Mobile Ad Hoc Networks", Proceeding of 26th Annual IEEE Conference on Local Computer Networks, 2001, 568-575.

[10] M.R.Pearlman, Z.J.Haas, et al, "On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks", Proceeding of Networking and Computing, 2000 Mobihoc, 3-10.

[11] S.J.Lee, M.Gerla, "Dynamic Load-Aware Routing in Ad Hoc Networks", Proceeding of IEEE International Conference on Communications, 2001 (ICC 2001), Vol.10, 3206-3210.

[12] J.Li, C.Blake, et al. "Capacity of ad hoc wireless networks". In ACM MobiCom, 2001.

Service control strategy of HSDPA based on Nash competition

Jun Qian, Yanping Yu, Kuang Wang, Guangxin Zhu
Department of information science and Electronic Engineering,
Information College, Zhejiang University, 310027
Email: 13505710057@zj139.com

ABSTRACT

Traditional packet scheduling strategies are based on the centralized control. Wireless multimedia services require faster computation capability and different individuality. It is an embarrassment to balance the save of unceasing investment and the satisfactions of Q&S. This paper presented the combination of Nash non-cooperate competition theory and the whole Q&S control framework in HSDPA service access. The participators chose maximum net utility independently and in distributed way based on the utility function matrix. The design and application of the presented packet scheduling strategy are simple and flexible compared with traditional ones. Simulation results show that the whole consumers' satisfaction and network operation safety of this new strategy improved distinctly and the fluctuation of network load by new strategies is smaller compared with that of fixed strategy and grade cost strategy.

Keywords: HSDPA, Nash-competition, service control strategy.

1. INTRUDUCTION

A new wireless access technology HSDPA named "High Speed Downlink Packet Access" grows rapidly whose wireless capacity is increased from 50 percent to 100 percent compared with Release 4 in WCDMA. But it has some shortcomings, such as poor pattern self-adjust ability, complex application environment, short of flexibility, which needs to predefine the transmission character to ensure wireless access etc. At the beginning of establishment it is difficult to maintain.

Many scholars make great efforts to realize high efficiency, simple control strategy. R. Love brought forward new control strategy based on the received quality of mobile terminals [1]. Elliot R.C [2] presented Max C/I algorithm. Asralan Farrokh proposed a strategy named Modified Largest Weighted Delay First control strategy [3,4]. Some specialists borrowed cycle Round Robin technology. Holtzman J.M presented asymptotic analysis of Proportional Fair algorithm (APF)[5]. Also Yang Liu got dynamic valve adjust (MTA_ISIR [6]). Kolding T presented radio aware Random Iterative Hybrid Scheduling technique (RIHS)[7]. Saied Abedi proposed the Environment Adaptive fast packet-scheduling algorithm with Single user Fairness guarantee [8](EASF) etc.

The services providers face with many new challenges, such as how to exactly adjust preferential unit price during different time of period in different new services. In recent years many scholars attempt some new concepts for reference from economy theory. At the beginning of establishing network there exist many blind or heavy load cells, which are difficult to maintain. Much manual interference can be saved if the concept of Nash non-cooperative competition in service control strategies is adopted which can be implemented easily with adaptability.

The rest of the paper is organized as follows. Firstly the Utility Function and Nash non-cooperation competition is introduced in section 2. Followed by section 2 there are many experiments, simulations and results of three different service control strategies named as fixed unit, graded unit and Nash non-cooperative competition in section 3. At last we get some conclusions in section 4.

2. UTILITY FUNCTION AND NASH NON-COOPERATIVE COMPETITION

We emphasize these service control strategies through controlling the transmitting power. Combining SIR with service parameter of different services, we conclude the subjective utility function and the unified consumers' satisfaction standard. At last we simulate the natural process of consumers' access, evaluate these different strategies.

2.1 Utility Function

In this paper $U_i(SIR)$ is defined as the degree of i th consumer's fondness to some services. The expression of the SIR and $U_i(SIR)$ model please refer to [8]**.** In multimedia services many different service grades are adopted for different data rate. Also the consumers' satisfaction changes with the different services grades provided by the system. The concept of network soft capacity is introduced naturally. The power transmission is proportional to the cost. The net utility of customer is the difference of the utility and cost function

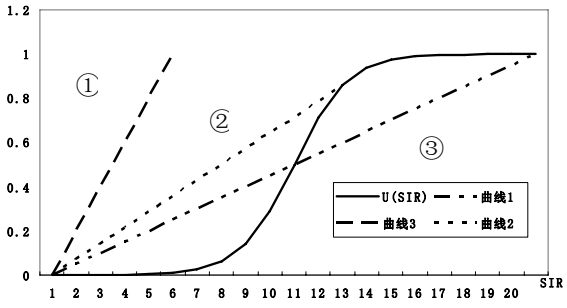


Fig. 1. The contrast between the utility function and cost function curve

The net utility varies as the cost curve slope changes. While the cost curve slope ③ is quite small there will be two points of intersection with utility function. With the increasment of the cost curve slope supers curve ② tangent with the utility function, the consumer will stop the transmission in order to escape from self-oscillation.

2.2 Nash non-cooperative Competition

Nash competition is a branch of Game Theory. Main research topics are focused on how to act with multi-competitors [9,10]. HSDPA based on utility maximum is an improvement in the development of service control compared with traditional centralized control.

Utility Matrix Definition: Suppose the participator 1 has m strategies could be chosen, the participator 2 has n strategies. The participator 1 select the ith strategy from the m strategies and the participators 2 select the jth strategy from the n strategies at the same time, i =1,2…m; j =1,2…n. The participators 1 and 2 get the utility a_{ij} and b_{ij}, respectively. The utility Matrixes are defined as $A=(a_{ij})_{m\times n}$, $B=(b_{ij})_{m\times n}$. In this paper, the services provider and the studied consumer are defined as the participator 1 and 2. The utility Matrix is denoted as follows:

Table 1. The Nash competitors' Utility Matrix

	The services provider		
The studied consumer		Network not busy	Network busy
	Access	（+3，+10）	（-3，-10）
	Wait or exit	（+1，-2）	（-1，+2）

In the above table,（+3，+10）represents that the studied consumer get the utility 3 and the services provider get the utility 10 when the customer try to access the network and the network is not busy. If the number is negative it means that the utility satisfaction of the studied consumer is poor or the service provider takes extra operation risk. The utility in different conditions can be deferred from the above table.

When the network is not busy, the services provider will provide cheaper unit price for promotion. And the consumers can get more net utility than usual. When the network is busy and the access consumers are willing to wait or exit, the consumer avoids the poor utility satisfaction and the service provider avoids the risk of the heavy load. But when consumers have strong willing to access the network despite the heavy load of network, the service provider has to increase the unit price, so the net utilities of the customers and the system become negative. Part of these consumers who are sensitive to the feedback would cut down their access services rank to avoid the high operating risk so that the utility receipted by the quality of high rank consumers can be guaranteed.

3 SIMULATION AND RESULTS

The simulation assumptions list as follows: There are 30 wandering consumers randomly distributed in a cell whose velocity is equal to 1 unit. The radius of the cell is 500 meters (about 1600 feet). The antenna height of the base station is 15 meters (about 50 feet). The maximum of the system power transmission is limited below 50 watts. The height of the consumers' terminals is supposed to be 1.5meters (5 feet). The intensity of background noise less than 10-6.The parameter *a* and *b* of utility function is 0.5 and 0(no difference between different service). The load of the network is limited to no more than ninety percent (90%). The guided unit price is 20. The maximum power output of each user is limited below 125% of usual. The services provider should stop the service while SIR is less than -12. The simulation adopts Monte Carlo method, which repeats 100 times. Three different services control strategies defined as fixed unit, graded unit and Nash competition are compared. With different velocity, controlled load and unit price, whole system net utility and power output stability of them are compared as follows.

3.1 The comparison of whole system net utility and the power output stability of the system under different random velocity

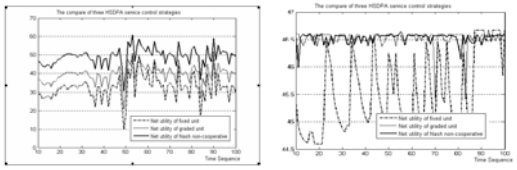


Fig. 2. The comparison of whole system net utility and the power output stability of the system while the consumers' random velocity are equal to 1, the network load is 90%, the unit price is 20

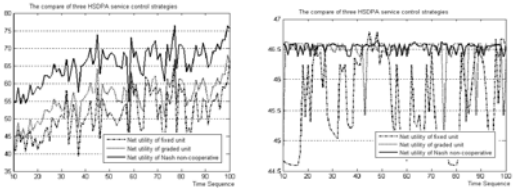


Fig. 3. The comparison while the consumers' random velocity is equal to 5

The wireless interference fluctuation ranges larger when the wandering velocity is higher (5 in this paper). The power output fluctuation of them is higher than usual. The whole net utility in Nash non-cooperative competition is better than the other two while the fixed unit one is the lowest. The power output fluctuation ranges in Nash non-cooperative competition is smaller than that of the other two.

3.2 The comparison under different network load control

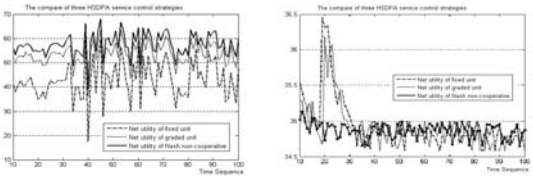


Fig. 4. The comparison while the network load is 70%

Fig. 5. (same as Fig. 2.) The comparison while the network load is 90%

From Fig. 4 the whole net utility in Nash non-cooperative competition is more than the other two while the fixed unit one is the lowest. Under the saturation of the network load (90%) the system needs more strict methods to control power so the whole output fluctuation range is less than that of the network load under 70%.

3.3 The comparison under different cost unit

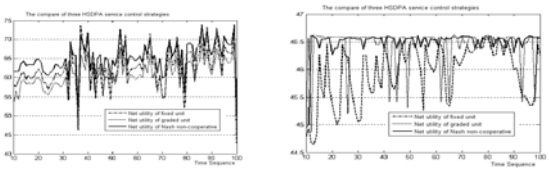


Fig. 6. The comparison while the unit price is 10

Fig. 7. (same as Fig. 2.) The comparison while the unit price is 20

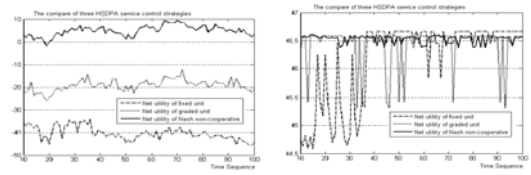


Fig. 8. The comparison while the unit price is 50

Unit price has direct influence on consumers' net utility of the whole system. The virtue based on the utility cannot be unfolded fully under low cost unit. The simulation result shows that the whole net utility under the three strategies are similar and the fluctuation range of whole net utility of fixed unit one is larger than other higher cost unit.

With the increase of the unit price up to 50 the consumers are more sensitive to the unit price, and need more strict methods to control the power output. The whole net utility in Nash non-cooperative competition is larger than that of the graded unit price while that of the fixed unit one is the lowest. The whole output fluctuation range of the three ones is less than the ones under cheaper unit price.

The Simulation results show that Nash non-cooperative competition in the service control strategy is better because of higher net utility and stable power output. It is an improvement in the development of services control that can be applied easily without signals interactive continuously [10]. Different access consumers chose different maximum net utility. The distributed service control doesn't need the central control.

At the beginning of the network operating a lot of operating risk can be avoided if we adopt it whose efficiency of the wireless access and the services quality can be improved by combining the different structure of cost and consumer needs, carefully evaluating wireless resource assignment.

4 CONCLUSIONS

The proposed service control strategy of HSDPA needs to be investigated further.

Further research can be done about the influence of utility curve slope variance on the service qualities produced and how to combine the price scheme and the various scheduling algorithms. And the satisfaction of all consumers will be unified. Researches can also be done in the area of extending existing Q&S protocols or mechanism to improve the performance.

On the other hand, the influence of the information asymmetry of Nash non-cooperative competition and the experience acquirement of repeated competition on the network is a research topic as well.

5 REFERENCES

[1] R. Love, A.Ghosh, et al, High Speed Downlink Packet Access Performance, VTC 2001, Spring, vol 3, 2234-2238.

[2] R.C Elliott, et al, Scheduling Algorithms for the CDMA 2000 Packet Data Evolution, Vehicular Technology Conference, 2002 VTC 2002 Fall, Volume1, 304-310.

[3] A. Farrokh, K. Vikram, Opportunistic Scheduling for Streaming Users in High-Speed Downlink Packet Access, IEEE Communications Society, Globecom 2004.

[4] A. Farrokh, V.Krishnamurthy, Opportunistic Scheduling for Real-Time Media Streaming in HSDPA Technology, in Submitted to IEEE Transactions on Multimedia, November 2004.

[5] Holtzman J.M, Asymptotic Analysis of the Proportional Fair Algorithm. ", Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium on, vol 2,September 2001,F-33-F-37.

[6] Yang Liu, Yu Kong Kwok, et al, A Practical Adaptive Packet Scheduling Algorithm with Single User Fairness Guarantee over the Forward Link of 3G Cellular Data Services, IEEE 2005,Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW05).

[7] Kolding T et al, Performance Aspects of WCDMA Systems with HSDPA ", Vehicular Technology Conference, 2002,VTC 2002 fall, vol 1,477- 481.

[8] M. Xiao, B. Ness, et al, A Utility-Based Power Control Scheme in Wireless Cellular Systems, IEEE/ACM Trans on Networks, vol 11(2), Apr 2003.

[9] E. Altaman, T. Basar, et al, Competitive routing in networks with polynomial cost, In Proceedings of the IEEE INFOCOM 2000,March 2000.

[10] T. Basar, G. J. Olsder, Dynamic Non-cooperative Game Theory ", 2nd Philadelphia, PA: SIAM, 1999.

IFSCM: Agent Based Supply Chain Management Integrative Framework

Chengxiang Yuan, Hua Hu, Yun Xu
College of Computer Science & Information Engineering, Zhejiang Gongshang University
Hanghzou, Zhejiang 310035, China
Email: vt00098@126.com, huhua@mail.zjgsu.edu.cn, xuyun@mail.hzic.edu.cn

ABSTRACT

Supply chain management (SCM), which needs the support of integrative information infrastructure, is critical to an enterprise. In this paper, an agent based integrative framework for SCM called IFSCM, is introduced. In IFSCM, multi-agent system is treated as organizations, then six models are used to describe the organizations, at last, an agent engine is designed to coordinate the cooperation of agents by the using of the description. IFSCM allows the quick integration of enterprise information infrastructure to effectively support the SCM business processes integration.

Keywords: software agent, supply chain management, integrative framework.

1. INTRODUCTION

The object of supply chain management is to make the best value for all the members of the chain[1]. It can enhance the overall competition of an enterprise.

SCM requires the quick building of information foundations that can support the supply chain members's business process integration in low price. Integrative technology is needed to abstract and encapsulate the member's information infrastructure into active software component in large or middle granularity in operation level. Then those active software components are used to construct high-level architecture[2]. Comparing to the integration of other applications, the supply chain of an enterprise is dynamic—members join and quit the chain continuously, and the information infrastructure of the member is complex. All these bring challenges to the Integrative technology.

Software component or middle ware based software is passive, in low abstract level, lack of the coordination ability. All these influence the coordination execution of software components, making it difficult to effectively support the rapid construction of SCM information infrastructure and execute in coordination.

Agent technology used today is too complex, making it difficult to support the rapid construction of SCM information infrastructure in low price and execute in coordination. In this paper, we advocate an integrative framework IFSCM, which uses model description to describe the coordination of agents. With the help of agent engine, those descriptions are used to control the coordination of agents.

Our goal is to develop a framework to support the rapid construction of SCM information infrastructure in low price and execution in coordination. The remainder of the paper is organized as follows. Section 2 outlines the system architecture and key technologies to be incorporated into the framework. Integration process is given in Section 3. We provide some preliminary results and clinical feedback in Section 4. Finally, some concluding remarks are presented in Section 5.

2. INTEGRATIVE FRAMEWORK IFSCM

The information foundations of supply chain members can be think of as applications that needed to be integrated dynamically when needed in a distributed environment. IFSCM focuses on the high level of the information foundations. It takes agents as a member of an organization that is obligated to carry out some kind of computing. With the help of organizational rules, agents can collaborate with each other. IFSCM supports the integration of computing activity and agents collaboration from two ways as follows, so as to support the rapid establishing and coordinated operation of integrated application system.

IFSCM =(MD，AE)

1) MD — — Model Description. The integrative application system is designed as a MAS that is described by six models: task model, describing the functions the MAS must implement; role model, describing the main roles and their obligation in the MAS; organization model, describing the roles each agent or organization plays; reaction model, describing how agents react with each other; belief model, describing how agents react with the environment; agent model, describing the characteristic of an agent.

2) AE—— Agent Engine. It uses the knowledge provided by the model description to manage the cooperation between agents and invoke the legacy software encapsulated in the agents when needed.

IFSCM takes the integration and coordination of the computing activity as the main thread for agents and MAS design and provide the method for integration and coordination of the computing activity.

2.1 TASK MODEL

In IFSCM, agents are used to encapsulate the legacy SCM applications. The computing carried out by the legacy software now becomes skills of agents. Task model describes the characteristic of all the tasks, such as, how to decompose it, how to schedule its subtasks and so on. Task that can be finished by a single agent is a task unit, denoted by "Task-Unit". Otherwise it's a compound task, denoted by "Compound-Task". Tasks can be defined in BNF as follows:

<Task> := [<Compound-Task>] | [<Task-Unit>]
<Compound-Task> := [<Name>][<Task>]⁺ [<Status-Mapping>] [<Input>][<Output>] [<PreProcessing>] [<ExceptionProcessing>][<EndProcessing>] [<Precondition>] [<PostCondition>][<SchedulingPlan>]
*<PreProcessing>:= [<function expression> | <rule> | <group of rules>]**
<ExceptionProcessing> := [<function expression > | <rule>

*|< group of rules >]**
<EndProcessing> := [<function expression > | <rule>
*| < group of rules >]**
*<Output> := [<Type> < variable name>]**
*<Input> := [<Type>< variable name >]**
<SchedulingPlan> := { <PlanStep>
*| (LOOP <SchedulingPlan>) }**
<PlanStep> := RETURN
| (← <TaskSet> [<Status>])
| (OR { (←<TaskSet> [<Status>]) }$^{+}$)
<Status-Mapping>:= $Status_1 \times Status_2 \times ... \times Status_n \rightarrow Status$ //$Status_i$ is the collection of subtask status
<Task-Unit> := [<Task-Name>][<Status>]$^{+}$[<Context>]$^{+}$[<Cs-Mapping>] [<Out-put>][<Input>][<PreProcessing>] [<ExceptionProcessing>][<EndProcessing>] [<PreCondition>][<PostCondition>] [<Constraint>]
<Context> := [<Data-Item>]$^{+}$
Cs-Mapping : = {(<Context> →< Status>)}$^{+}$

PreProcessing represents some other tasks must be done before the task we defined. The status space of a Compound-Task is determined by the status space of its subtasks. Context here means a snapshot of some data important to the task. The execution of the subtasks must be scheduled. Static schedule is used here. For a Compound task, we define many steps, one of which is selected to execute according to the task's status. Symbols used here such as →(total function) is observed to the rule of Z language.

2.2 ROLE MODEL

Agents play different roles when cooperating. The role is an abstract representation of an agent function, service or identification[3] within an organization. A role declares the obligations members must be charged with when they play it. Role model can be defined as follows:

Role := [<Role-Name>][<Task-Name>]$^{+}$

2.3 ORGANIZATION MODEL

Agents can be separated into organizations. An organization consisting of a group of agents, is supposed to perform some computing activities. It has some kind of organizational structure, observing some kind of organizational rules. Its members are agents or sub-organizations. Organizational Rule expresses general, global requirements for the proper instantiation and execution of MAS. An organizational structure defines the specific class of organization and control regime to which the agents and roles have to conform in order for the whole MAS to work efficiently and according to its specified requirements[4]. Organizational patterns express pre-defined and ubiquitous organizational structures that can be re-used from system to system. Organization is defined as follows:

<Organization>:= [<Organization-Name>][<Member>]$^{+}$ [<Organization-Rule>]*[<Role-Name>]*
<Member> := Organization | Agent
Organization-Rule: {(< Role-Name> →<Member>)}$^{+}$

There are many potential Organization structure, only few of them are frequently used[4,5]. Organizational patterns can be obtained when these organization structures are classified and abstracted. Organization research can make it possible for us to choose the best suitable method to coordinate the agents.

2.4 COOPERATION MODEL

Agents need to collaborate with each other. There is information flow and control flow among them. A sequence of messages among agents is called a conversation. There are four types of conversation: command, inform, query and negotiate. Cooperation model can be defined as follows:

<Conversation> := [<Conversation-Type>] [<Participator-Conversation-Content>]$^{+}$
<Conversation-Type> := "Ask"|"Inform"|"Query"|"Negotiate"
<Participator-Conversation-Content> := [<Participator-Role>][<Conversation-Content>]
<Conversation-Content> := [<PreProcessing>] [<EndProcessing>][<Context>]$^{+}$[<Con-versation-Status>]$^{+}$ [<Cs-Mapping>]$^{+}$[<Event>]$^{+}$[<Translation>]$^{+}$ [<a-operator>]$^{+}$[<Constraint>]$^{+}$
<Event> := [<Conversation-Verb>][<Event-Name>] [<Data-Item>]$^{+}$
<Conversation-Verb> := "Ask"|"Inform"|"Query" |"Negotiate"|"Reply"
<Translation> : ((Conversation-Status → a-operator) → Conversation-Status) | (Conversation-Status, Event. Event-Name) →Conversation-Status

Conversation can be defined by type and content. Conversation content is defined for each role participating in the conversation. Constraint is the limit must be observed, for example, the time a conversation can last. There are five conversation verbs. The verb used by the first event of a conversation identifies the type of the conversation. All the other verbs used by the conversation are "Reply".

2.5 BELIEF MODEL

Belief model describes how agents react with the environment, when to launch a conversation, when to carry out a task and so on.

<Belief> := Dest-Action-Namet ← (Source-Action-Name, Context)
<Dest-Action-Name> := Action-Name
<Source-Action-Name> := Action-Name
<Action-Name> := Conversation-Type | Task-Name

When a task or a conversation is finished, other tasks or conversations may be triggered according to the context, which is defined by belief model.

2.6 AGENT MODEL

Agent model is used to describe the characteristic of an agent, such as agent identification, services can provide, ontologies used, communication method and so on. Agents can be defined as follows:

<Agent> := [<Agent-Name>][<Service-Name>]$^{+}$ [<Ontology>]$^{+}$[<Communication-Method>]$^{+}$ [(<Task-Unit><Function>)]$^{+}$

Given the model description, an agent is transparence to others. So other agents can communicate with it, schedule it to participate in a Compound-Task. The element "[(<Task-Unit><Function>)]" tells the agent how to schedule a function to finish a task.

Six models are used to describe a MAS. These descriptions are used to control the collaboration among agents. To make use of the descriptions, an agent engine is designed.

2.7 AGENT ENGINE AE

An implement of an agent can be divided into four parts: collection of methods, inference engine, cooperation engine, and execution engine, just as figure 1 shows.

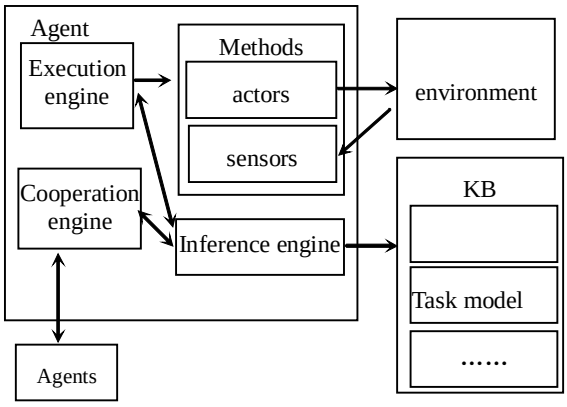


Fig. 1. structure of agent

2.8 AGENT ENGINE AE

An implement of an agent can be divided into four parts: collection of methods, inference engine, cooperation engine, and execution engine, just as figure 1 shows.

1) Collection of methods

After being encapsulated into an agent, legacy software becomes a method of this agent. As a sensor or reactor of an agent, it senses the environment and reacts with the environment.

2) Inference engine

Inference engine is used to access the KB and to infer. As the control center of an agent, it coordinates the execution of execution engine and cooperation engine.

3) Cooperation engine

Cooperation engine handles the conversation between agents. It takes charge of message handling, coordinating multi-conversations going along simultaneity, estimating whether a conversation constraint is met and validating the conversation.

4) Execution engine

Execution engine is supposed to invoke the methods to fulfill some tasks. If an agent is an undertaker of a compound task, it takes charge of coordinating the execution of the compound task's subtasks.

For the four parts, the collection of methods is domain dependent. The others are domain independent. These domain independent parts can be used to handle the coordination between agents. Agent engine consists of those three parts. IFSCM-based integrative information infrastructure consists of agents and knowledge base. The model definition can be viewed as a MAS description ontology, which will be instanced to form a knowledge base (KB) when information infrastructure being integrated. The KB then is used to control the coordination between agents. An agent can be scheduled by other agents. Before carrying out a task, agent gets the description of the task from the KB. If the task is a compound task, it will schedule other agents to fulfill it according to the organizational model and role model. If the task is a task-unit, agent finishes it according to the agent model. Belief model is used to determine whether a task or a conversion is needed to be carried out when a task or is completed.

3. INTEGRATIVE PROCESS

The integrative processes consist of two steps: in the first step, inter-enterprise applications are integrated. Then the second step are intra-enterprise. For the step one, we analyze which function is needed to be integrated and encapsulate the legacy software and build the task model. Referring to some kind of organizational pattern, the encapsulated software can be aggregated into roles. Those roles then are aggregated into agents. So the role, agent and organization model are given. According to the information flow and control flow among agents, cooperation model is created. At last, we analyze how task and conversation are triggered to get the belief model. As it can be seen, IFSCM based integration focuses mainly on legacy software encapsulation and model description creation.

For the intra-enterprise integration, we only need to create the model description, because all the legacy software has been encapsulated in the first step.

4. APPLICATION

So far many details about IFSCM are given. Now we use the IFSCM to simulate the integration of information infrastructure of Tezhou Company, a subsidiary company of Wangxiang Group in Hangzhou. In Tezhou, we are only interesting in four departments: project, manufacture, market and order department. The market department receives orders from other companies. The project department makes plans for the order. Those plans are sent to manufacture and order department. The manufacture department so can make its daily working plan. The order department gives orders for the material. We use a MAS with four agents to fulfill those function, just as figure 2 shows. When a market agent receives orders from other company, it handles the order and makes a production plan then sends them to the plan agent.

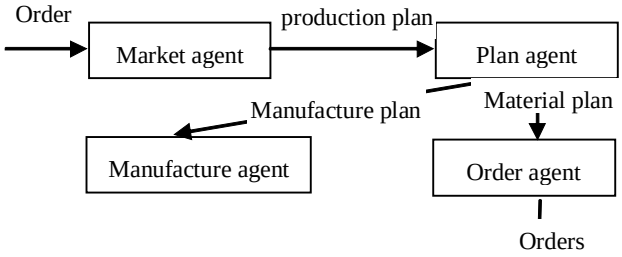


Fig. 2. organization of agents

The later makes a manufacture plan and material plan for those productions and sends them to manufacture agent and order agent respectively. After receiving that, the manufacture agent and order agent can make their own plans. All those can be done automatically without the intervention of people. For the limitation of this paper, we don't want to give the detail of how to describe the models. We only illustrate how agent can collaborate with each other.

5. SUMMARY

This paper proposes an agent based integrative framework IFSCM. From two aspects: model description and agent engine to support the rapid construction of SCM information infrastructure and execution in coordination.

6. REFERENCE

[1] Andrew Cox. Power, "Value and Supply Chain

Management". Supply Chain Management: An International Journal, Vol 4 :192-198.

[2] J.Gao, D.H.Lin. "Agent base virtual enterprise integrative framework IFVO". Journal of Computer Research and Development, 36(12) ,Dec. 1999.

[3] J. Ferber， O. Gutknecht. "A meta-model for the analysis and design of organizations in multi-Agent systems". In Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS 98). IEEE CS Press, June 1998.

[4] Zambonelli, F. Jennings, N.R.Wooldridge, M (2001): "Organisational Rules as an Abstraction for the Analysis and Design of Multi-Agent Systems". In: Int J. of Software Engineering and Knowledge Engineering 11(3) :303-328.

[5] M.S.Fox. "An organizational view of distributed systems". IEEE Transactions on Systems, Man, and Cybernetics, ,January 1981, 11(1):70-80

Research of Download Model based on Multi-Agent Collaboration

Chengcai Mei , Huafeng Guo
Department of Information and Engineering, Zhejiang Industry & Trade Polytechnic
Wenzhou Zhejiang, 325003, P.R.China
Email: breakout2008@sina.com

ABSTRACT

At present, many download tools can help us accelerate download speed. Nevertheless, their download speed is limited by the bandwidth connecting to the target website and the target website's restriction on the number of connecting threads of single IP. This paper presents a download model based on multi-agent collaboration to overcome the deficiency of current download tools. According to some rules, the model dispatched download tasks to agents on network. Through the collaboration of these agents, download tasks can be accomplished rapidly, efficiently and flexibly. The mode has two advantages. First, download tasks are completed by several agents, and then the data is sent to the request agent. So there are no restrictions on the maximum number of connecting threads of single IP. Second, download tasks are completed by several agents, so failure in network connection will have little influence on the whole task.

Keywords: Agent, Download, Collaboration, P2P, FlashGet.

1. INTRODUCTION

Download is a major way of getting resources from Internet because of its quickness. Now there are many practical download tools with powerful functions, such as NetAnt, FlashGet, CuteFtp, BitTorrent and so on. Most of these tools support multithread download and breakpoint download, which make the tools more efficient and applicable. However, these tools also have some disadvantages: (1) in spite of their support of multithread download, the working efficiency of NetAnt, FlashGet, CuteFtp is limited by the bandwidth connecting to the target website and their applicability is also limited by the target website's restriction on the number of connecting threads of single IP; (2) BitTorrent [1] and its colleagues take a new point-to-point download, so they do not face the limitations other existing download tools face. However, BitTorrent is efficient only when more people download the same resource at the same time and when the seed is 0, the download may be incomplete. Besides, BitTorrent can only be used to download the resources on the computers, which also run BitTorrent.

To solve the problem of the download tools mentioned above, this paper presents a download model based on multi-agent cooperation, which will make download task quicker and more efficient.

2. MULTI-AGENT TECHNOLOGY

Multi-agent [2] origins from distributed artificial intelligence, and are usually considered to be able to run automatically under certain circumstances and to be autonomous, responsive, cooperative and self-learning. The ability of single agent is limited; multi-agent system [3] is a self-organizing system composed of many autonomous agents based on certain cooperative mechanism and it aims to solving problems that beyond the single agent. Cooperation among multi-agents, task decomposition and task coordination of a multi-agent system can improve the system's ability, and cooperation among multi-agents can overcome the problems in a single agent such as the incomplete knowledge and uncertain information. Multi-agent system can be intensively organized and can also be loosely organized and its granularity is flexible. A distributed problem solving system is a distributed agent network usually composed of a group of incompact coupling agents, in which every agent can complete part of a complex problem solving and can change its behavior according to environment and can also communicate and cooperate with other agents.

3. SYSTEM MODEL

In a multi-agent system, agent can form a union by negotiation to accomplish task solving [4]. The symbiotic module of this system is based on contract net protocol, in which agents are divided into two types: manager and executor. The managers send information to executors and deal with the result; executors are responsible for specific tasks. In some cases, an agent can play different roles on different stages [5]. The advantages of contract net lie in its simple structure, good extension; the disadvantages are low efficiency and large cost of broadcasting communication [6]. So we present an improved contract net model, called friend agent symbiotic model, which is composed of a dispatch agent and several download agents (as Fig. 1. shows). Download agents can play two different roles: request agent and server agent. Request agents are the agents that send download requests; server agent is the agent that responds to the download requests from the request agents and offers them corresponding services. The server agent that has good cooperation with request agent can be defined as its friend agent. Every download agent has certain inferential capability and its own knowledge base. Usually, download agent, as independent software, can complete the download task given by users. Dispatch agent is the dispatch centre of download agents, and its task is to maintain the basic information of all online download agents, including their IDs, net speed, available hard disk space, available network bandwidth, current service state, credit level and so on. When a download agent puts in a download request, dispatch agent will search for server agents and update its information.

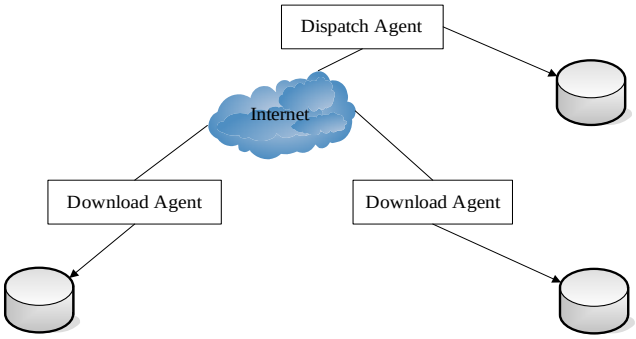


Fig. 1. The system architecture

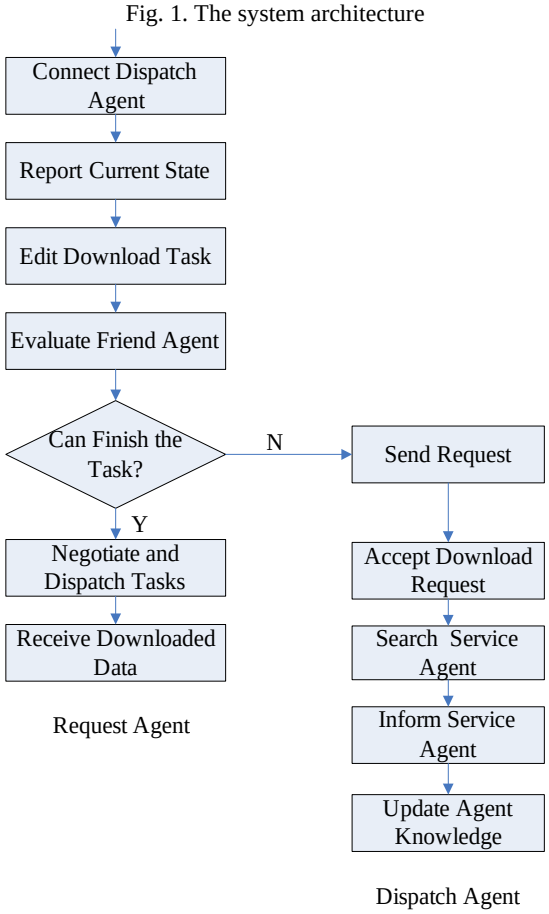


Fig. 2 .The model cooperation mechanism

The cooperation mechanism of the model is as Fig. 2. Shows. When a download agent is online, it sends its state to the dispatch agent, and the dispatch agent will update download agent's information. If a download agent wants to download some resources, it will give a download task list, which includes the addresses of the resources, the importance rank of the download task and the deadline of the download task. Then it becomes a request agent. According to the knowledge in knowledge base, request agents first give an evaluation on the conditions necessary for the completion of the download task, including network bandwidth, online time, hard disk space, etc. Then it will seek for its friend agents in knowledge base and negotiate with them and next, gives an evaluation on the friend agents' ability. If the existing friend agents can complete the task, it will assign the task to them; if they can't, it will send request to dispatch agent for help from other download agents. Then the dispatch agent will seek for suitable server agent and inform it. With the information informed, the server agent first gives an evaluation on the download task, if it is competent of the task, it will communicate with the request agent, inform it the current service state, begin the download task, and send the downloaded data to request agent. After receiving the data, request agent will update its knowledge base.

4. IMPLEMENTATION OF THE MODEL

4.1. IMPLEMENTATION OF AGENT SERVICE LOGIC AND SELF-LEARNING

On receiving a cooperation request, server agent will analyze the request information, which includes: request agent's ID, network site, credit rank, time that request is sent, website list of resources to be downloaded and the deadline of every download task. Server agent will judge whether:

(1) Data size of the download task < maximum available hard disk space

(2) Request agent's credit > prescribed credit rank

(3) Whether the request agent's access right is within the system's security clearance;

(4) Min {(capability of the download task / connection speed to the target website)*a1, the average networking time in the current period}+(capability of the download task / the connection speed between the request agent and server agent)*a2< the time limitation of the completion of the download task.

Among them, a1 and a2 are the risk coefficient set in case of a failure connection, and a1>1, a2>1.

If all the above conditions are met, it means that the task can be completed as anticipated. Then the service agent will inform the request agent to agree cooperation and update knowledge base according to the relevant content of this cooperation. It will begin the download task and send the downloaded resources to the request agent after download is finished. As Fig. 3. Shows.

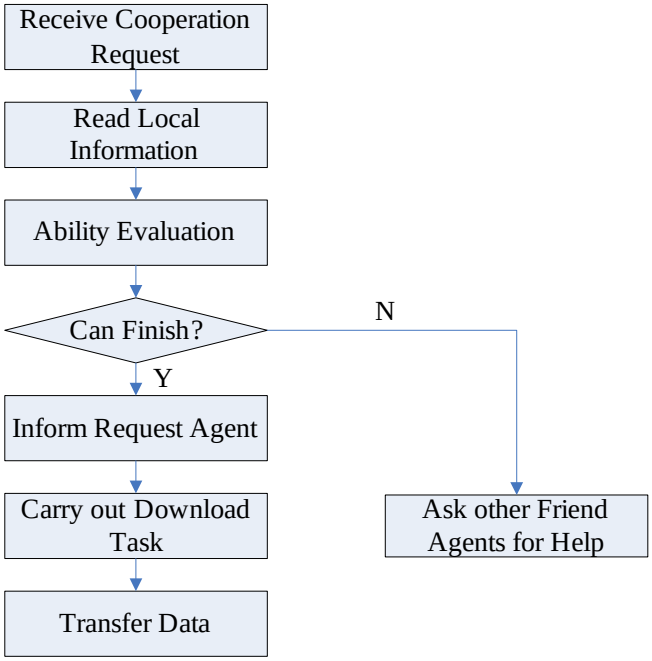


Fig. 3. Service logic and self-learning flow of the agent

4.2 COMMON ERRORS AND SOLUTIONS

Failure in network connection and users' human factors may result in a failure in the multi-agent cooperation. The key

point in completing the download task successfully is to prevent and remove these errors. Common errors are:

(1) The network state of the service agent changes during downloading process (such as the network connection speed slows), which make the task cannot be completed as anticipated. For this error, the system gives the following solutions:

First, the time of completing task will be divided into several time quantum and in every time quantum service agent will report its current state to request agent, which will decide whether the agent should be in service: if the service agent can not finish the task as anticipated, the request agent will resort to other friend agents for help or have the service time prolonged;

Second, if the server agent can not finish the task as anticipated, it will ask its friend agents for help to accomplish the task together.

(2) The interference of human factors influences the normal cooperation between the agents. For example, a server agent user gives too much download tasks and refuses service for other agents, or reduces open network bandwidth or available hard disk space. System will solve the problem by credit rank. Credit rank means the ability of the agent to serve other download agents and the number of service it provides. The more it serves other agents, the higher its credit rank is. If a user provides fewer resources for a long time than before, which make the task can not be completed as anticipated, system will reduce his credit rank as a punishment, and at the same time, it may be put into black list by the request agent. Besides, with low credit rank, the user can only submit task of smaller size, and it will get fewer cooperative agents.

(3) Service agent finishes the download task but can not send the downloaded resources to download agent because of connection failure between download agent and service agent. To solve this problem, the system adopts a multi-agent cooperation based on mailboxes, which takes mailboxes as temporary data storage space in case of connection failure. So request agent can get the data from mailboxes.

Several agents to keep certain redundancy will complete important tasks, so failure in one service agent will have no effect on the completion of the download task. This is also available to (1), (2) and (3).

4.3 EFFICIENCY IMPROVING

As we can see from the whole download process of this system, time is mainly spend on: dispatch agent's searching for the service agent, communication between request agent and service agent, service agent's completing the task, service agent's sending the data to request agent. Comparing with traditional download tools, this system spend more time on: dispatch agent's searching for the service agent, communication between request agent and service agent, service agent's sending the data to request agent. The research indicates that when the download task is of a small size, the benefit with multi-agent download may not be able to make up the spending of the time of the communication and negotiation between agents. So the system gives limitation to every download task's size of the download agent. Every download agent gives its own minimum threshold of single download task. And the agent itself will complete tasks of a smaller size.

In addition, when there are too many agents requesting download, the dispatch agent will spend much time on searching for service agents. To avoid being the bottleneck of the system and to make logic less complex, the dispatch agent is assigned only in charge of the most fundamental searching task, which is to search for the disengaged service agents and have the searching algorithm optimized. At the same time, the dispatch agent takes a queue mechanism for the requests and serves request agents according to "First Come First Serve" and their credit rank. Download agents keep their friend agents' information and they can cooperate with each other without dispatch agent's interference. In this way, dispatch agent's burden will be reduced a lot. It also takes time for service agent to send downloaded data to request agent. So in choosing service agents, request agent should try to choose the one that can complete the task with less time.

5. APPLICATION EXAMPLES

Based on the model mentioned above, we implement a download system based on multi-agent with a dispatch agent and five download agents. When there are download tasks, dispatch agent will dispatch the tasks to download agents according to the size of the tasks and agents' states. We had given a test to this system on the Campus-net. The test was set at 18:00 ~ 20:00, which is the peak-hour of the Campus-net. The system works well and the results are as Table 1. Shows.

The test results show that, with agent cooperation, the system's efficiency is twice of that of traditional download tools, such as FlashGet, and the bigger the size of the task is, the more efficiently it works.

Table 1. Test results
(All the data is average value of many tests)

File Size/ MB	Working Agent Amount at the same time	Download Time of Agent Cooperation/ S	Download Time of FlashGet / S
20	3	29	41
20	5	21	41
100	3	133	215
100	5	98	215
400	3	578	960
400	5	364	960

6. CONCLUSIONS

The download model based on multi-agent cooperation has the following advantages:

(1) The download tasks are completed by several agents, and then the data is sent to the request agent. So there are no restrictions on the maximum number of connecting threads of single IP. This is particularly available when the request agent and the service agent are with the same highspeed network (such as LAN).

(2) The task is completed by several agents, so failure in network connection will have no influence on the whole task. Of course we may have a long way to go before this system can be put into actual application. There are many problems to be studied and solved.

7. REFERENCES

[1] H. Zhang, W. Li, Study of Big Network File Distribution Technology [J]. Micromachine and Application. 2004, 39(7): 28-30.

[2] H. Li, Q. Wu, Summary on Research of Multi-agent System. Journal of Tongji University. 2003, 31(6): 728-731.
[3] Y. Fan, J. Cao, Theory, method and application of Multi-Agent system [M]. Beijing: Tsinghua University Press,2002
[4] F.L. Zhao, A cooperative Agent modeling approach for process planning [J]. Computer in Industry, 2000,41(1): 83-97.
[5] B. Zhao, Y. Fan, Applications of MAS to production scheduling research [J]. Control and Decision, 2003,18(1): 1-6.
[6] M, Cai, L, Lin et al, Multi-Agent-Based CAPP System. Journal of Computer-aided Design & Computer Graphics [J]. 2003,15(1): 96-101.

Research ON Multi-Agent Based Organization Decision Support System

Guozheng WANG
College of Computer & Information Engineering, Zhejiang Gongshang University
Email: wgz@ mail.zjgsu.edu.cn
And
Chunhua JU
149 Jiaogong Rd., Hangzhou, Zhejiang, China 310035
Email: guchunhua@hotmail.com

ABSTRACT

With the development of Agent, the technology of multi-agent begins to be applied in cooperative decision and solving problems. This paper presented the architecture and composition of ODSS-HY that is based on interactive collaboration of agents. The architecture contains organizer agent, processing agents, and communication agent. The application of ODSS-HY in one large electric appliance company is also introduced in this paper.

Keywords: Organization, Multi-agent, DSS, Model base.

1. INTRODUCTION

Since 1970s, with the development of AI, high-rank managerial and decision-making administrators confront with many significant decision-making problems such as strategy planning. These problems reflect the comprehensive results of their scheme, touch upon large scales and have many changing factors. In this situation, DSS develops rapidly. But many large-scale managerial decision activities are not convenient and even impossible to be carried out in centralized form, so researchers attempted to make decision at higher decision-making level and in more complicated environment that supported by computer technology. Therefore, high-rank DSS (called DDSS or ODSS) [1] came into being.

As many sophisticated systems are distributed, distributed intelligence systems attract more and more attention, and have promising prospect of large application, such as multi-intelligent robots cooperation [2], multi-expert system collaboration [3], concurrent engineering [4], man-machine collaboration [5] and communications region [6] Multi-Agent (MA) system is one of the main research branches, which includes the study of the knowledge, goal, planning, and the way of harmonizing the behavior of MA, namely the mechanism of collaboration and controlling. From the respect of the science of managerial decision-making and systems engineering, the paper adds Agent to ODSS to enhance the ability of assistant decision-making.

ODSS is the combination of one host computer and communication technology. The aim of ODSS is to harmonize and transit agreement on the decision design among functional modules and hierarchies [7]. ODSS makes use of computer and communication technology to enhance the processing of organization decision; it supports the decision-maker to carry out organization decision under friendly interactive environment by providing common data, data within the organization range and models. The features of ODSS are as follows [8]:

Relating to both common data and private data, and including both common models and private models;

One ODSS can cover many organization departments;

Greatly support the content (namely helping the decision-maker to carry out problem analyses by offering analytical tools), and also support the procedure (namely establishing the environment for decision analysis, supporting the decision-maker to accomplish all the activities in the decision-making procedure);

Breaking functional field hierarchical arrangement;

Having one set of tool kit for supporting information and processing task relying on computer technology.

The research of Multi-Agents refers to many problems such as how to harmonize the intelligent behaviors among one set of independent Agents, how to harmonize their knowledge, targets, and planning so as to get united to take action or to solve problems [9]. In this situation, though one Agent can be the exerciser of a certain task, it has open interface that can be accessed by any other Agent exercisers. One agent is able to deal with single target as well as multi-targets. From the angle of knowledge manipulation, intelligent Agent is a dynamic unit that can make use of its given knowledge to solve problems [10].

Many large-scale managerial decision activities are impossible or inconvenient to be carried out in centralized form. Most of the activities involve many organization units and decision-makers that undertake different responsibilities. In the large activity area, there are some significant decision-making elements and information resources that are needed in the decision-making procedure. So this paper combined MA with ODSS to solve the problem.

In addition, the goal of ODSS is to deal with the semi-structured strategic decision-making problems, which need several decision-makers to cooperate to solve problems. Besides, the decision problem itself might distribute in function, geography or time. And the decision-makers might locate in different places as well. MA is the right tool to solve the complicated problems that might distribute in function or geography and to figure out some debatable ones that emerge during inference (e.g. messaging). At the same time, MA speeds up the problem solving and adds security (including uncertain data and compatible knowledge), and extensibility (the number of the Agents and changing messaging). Meanwhile, the cooperation strategies, the raising of the collision and counteraction strategies of MA are also used by ODSS for reference.

Above all, by adding Agent technology to ODSS, the paper designed and developed Organization Decision Support System (ODSS-HY) based on MA.

2. THE ARCHITECTURE OF ODSS-HY SYSTEM

Based on the original integrated information system, HY-CIMS provides software platform, hardware platform, and data system. ODSS-HY sets up model base, agents, and decision data system to realize the organization decision

support for the whole enterprise. In our system, the open system is adopted. The architecture of the system is shown as Fig. 1.:

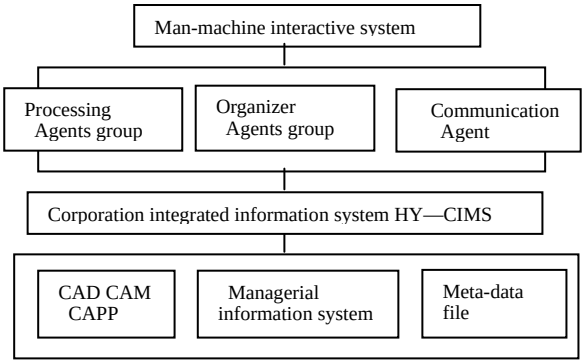


Fig. 1. The architecture of ODSS—HY

The system is made up of organizer Agent, processing Agents group, information Agent. These three Agents have independent functions, and interact with each other to solve problems concurrently through the organization and coordination of organizer Agent.

2.1 Organization Agent

Organization Agent, as the core to solve decision problems, is in charge of problem solving, distribution, cooperation, monitoring and recording the decision process and the confirmation of the result of the decision. It contains blackboard, inference engine, knowledge system and interactive interface.

Blackboard is an aggregation that records the information transition, task request and implementation procedure during problem solving. Blackboard has the following features:

The content of the blackboard could represent the current state of problem solving;

Blackboard is available to every Agent. Any Agent that takes part in the procedure of problem solving could output its conclusion to it.

Inference engine (IE), as the core of intelligent Agent, is the system for inference and problem solving. It takes charge of the interactive cooperation among Agents and the initial choice of plan execution.

Knowledge system (KS) represents the used model and the description of the experiential knowledge during the problem solving of the Agents. It represents the interactive behavior and provides grammar and semantics.

Interactive interface, as a handle based on KS, is not only responsible for the communication interactive processing, but also the interpretation and explanation of all kinds of models and description of experiential knowledge.

2.2 Processing Agent Block

Processing Agent Group consists of several processing Agents to handle the decision tasks; the Group realizes the integration of data information, models and knowledge and come to the settlement of decision problem and the conclusion. One single processing Agent is made up of Knowledge System and Problem Processing System. KS contains knowledge base, database, model base and the management system. Problem Processing System has problem analytical subsystem and processing subsystem. Its architecture is as Fig. 2:

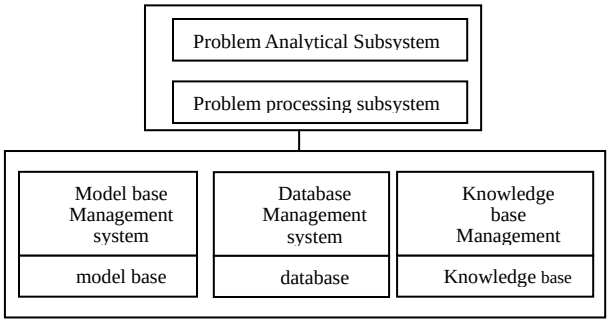


Fig. 2 Basic structure of processing Agent

The main work of analytical subsystem is to nail down the decision problem including the object, character, standard and controlling ingredients and to confirm the tunable models and knowledge. Problem processing subsystem is to carry out problem processing based on model, knowledge, data and method after problem analyses. During the processing, it interacts with organizer Agent to carry on, hand in and transit the task. The procedure, parameter, and conclusion of processing subsystem are all recorded on the blackboard till the end of the decision.

2.3 Communication Agent

Communication Agent supports the communication and cooperation among Agents that are mainly the single processing Agent in organizer Agents and processing Agent group. It is made up of communication consultation layer and communication rule governing layer.

Communication consultation layer chooses appropriate, efficient and skillful processing Agent to carry out problem processing. It creates a dynamic multi-Agent system that is better to answer the requirement of different decision problems. Knowledge system includes the knowledge about the ability, character, and state of information, adaptability of the Agents. During its communication, communication Agent needs to search KS continuously so as to select the appropriate Agents for the current decision problem.

Communication rule governing layer is mainly to set the communication standards for Agents. It harmonizes the processing activities of the Agents and makes sure that the activities do not deviate, and the harmony and cooperation among Agents are enhanced. Communication rule governing layer interacts with the rule base to perform its function of coordination and governing. The rule base is to: prescribe the junction with which the Agent could communicate; the content of the communication; the priority level; and the requirement of punctuation.

3. APPLICATION OF ODSS-HY

ODSS-HY is applied in Hangzhou Hongyan Electrical Company. This paper takes the design of one switch as an example to give a brief summary of the application of ODSS-HY.

In the real application of ODSS-HY, project team sets the chief engineer office as the organizer Agent responsible for distributing, decomposing, supervision and harmonizing, and establishes designing office, technical office and quality controlling office at the same time. The relation is as shown in Fig. 3..

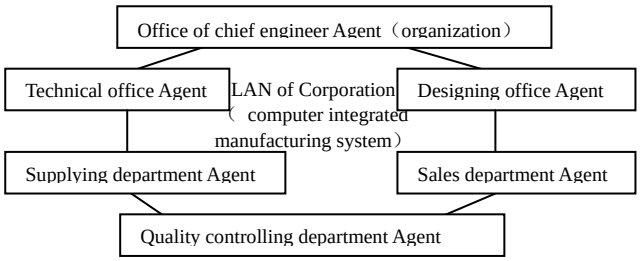


Fig. 3. Relation among agents in product design

In processing the problem of product design, the processing Agents make use of its own capability and problem solving system to handle the task. The result is the basis for collaborative processing, and the results of the processing by each Agent usually collide with each other. The reasons are as follow:

- Due to the limitation of the agents' knowledge and ability, the agents could finish limited task;
- Every Agent has their own knowledge backgrounds that are knowledge base, model base, database, so they attach importance to different points;
- Though ODSS-HY attempts to perfect its equipment, the processing information that each Agent got does not all complete which cause the reorganization of the processing environment and the problem itself deviates. Therefore, based on the independent processing and with the help of processing Agent, each Agent must cooperate to perform the problem processing which is arranged by organizer Agent. During the processing, with the help of the Blackboard, each processing Agent studies the results that other Agents have got. Through communication Agent and the blackboard of the organizer Agent, the Agents communicated with each other to get deeper awareness of the problem. Then, each Agent again asks the organizer Agent for the processing task. According to the former processing, organizer Agent furthered the distribution of the task. Each processing Agent performs the processing for the second time and then recorded the results, methods and explanation of the processing on the blackboard. All participant Agents could read the information and discuss, then, modify their processing results.

Eventually, under the coordination of organizer Agents, if every Agent reaches agreement, the result of the consultation, which is, sorted out by organizer Agent is the final result of the processing. If the collision still exists, the above procedure must be carried on repeatedly till the problem is solved.

4. SUMMARY

Based on the research of ODSS and MA, this paper designs and develops ODSS-HY that is applied to the CIMS project in Hangzhou Hongyan Electrical Company. But the above-mentioned example is just a simple product design. The fact would be more complicated. In the future we will aim at the more practical ODSS for distributed application in the organization-wide network.

5. REFERENCES

[1] E.B.Swanson, "Distributed decision support systems: A perspective", In: Proc.23rd Annual Hawaii Int. Conf. on System Sciences, 1990, (3): 129-136.

[2] C. H. Ju, Y. Ling, et al, "Agent-Based and Software Composite DDSS", 36th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-Asia'00), 2000, 50-57.

[3] Robert T. Sumichrast, "A Multi-Expert System for Material Cutting Plan Generation", Expert Systems with Applications, 2000, 19(1): 19-29.

[4] A.P. Ansuj, C.R. dos Santos, et al, "Concurrent Engineering Tools for Product and Process Improvement in an Electronic Industry", Modelling and Simulation, 2003, pp.380-388.

[5] C. J. Huang, M. C. Liu, et al, "Application of Machine Learning Techniques to Web-Based Intelligent Learning Diagnosis System", Fourth International Conference on Hybrid Intelligent Systems (HIS'04), 2004, 242-247.

[6] X. T. Gou, W. D. Jin, "Multi-Agent System for Multimedia Communications Traversing NAT/Firewall in Next Generation Networks", Second Annual Conference on Communication Networks and Services Research (CNSR'04), 2004, 99-104.

[7] H M Chung, R. Malapert, "Distributed decision support system: Characterization and Design Choices", Proc. 26th Annual Hawaii Int. Conf. on System Sciences, 1993, 660--667.

[8] A. K. Aggarwal, R. Mirani, "Policy Implications Of Organizational Decision Support Systems", the 1996 Hawaii International Conference on System Sciences (HICSS-29), 1996, 48-53.

[9] A Rao, M.Georeff, "BDI agents: From theory to practice", In proc. Of the First Int. Conf. on Multi—Agent Systems, 1995, 312—319.

[10] M.J. Wooldridge, N.R. Jennings, "Intelligent agents: Theory and practice", The Knowledge Engineering Review, 1995, 10(2): 115-1.

Multi-Agent System for Enterprise Application Integration Based on Aspect Oriented Technology

Xiaojun Li
College of Computer and Information Engineering, Zhejiang Gongshang University,
Hangzhou, Zhejiang, 310035, P.R.China
Email：lixj@mail.zjgsu.edu.cn
Aqun Deng
College of Material and Chemical Engineering, Zhejiang University,
Hangzhou, Zhejiang, 310027, P.R.China
Email：deng-aqun@hzfuji.com

ABSTRACT

The idea of enterprise application integration (EAI) was introduced, which is a popular issue in the IT to solve the problems regarding the information island in enterprise. Because the multi-intelligent system (MAS) possesses the advantages such as cooperatively solving problems, describing problem naturally, and using currently existing software by integration, it is convenient to provide new solutions to enterprise application integration. In view of the above understanding, the method introducing the multi-intelligent technology to the research upon the enterprise application integration was proposed. In general, the implementation of agent is to view the agent as an object, which encapsulates behaviors and states together. However, from the point of actual process of development, this method still has some shortcomings. Therefore, aspect-oriented technology was adopted to construct the model of agent and implement the multi-agent system. This approach may overcome the shortcomings of object-oriented method by using the advantage of aspect in design and programming.

Keywords: Enterprise Application Integration, Multi-Agent System, Aspect Oriented Technology, AOP.

1. INTRODUCTION

With the development of the computer software technology and the enterprise informationization, more and more various management information systems such as finance management system, production management system and sale management system are used in enterprise. But existing systems are generally applied in special function section and manage different object, and each system runs independently and lacks effective communication because of history. As a result, "the isolated information islands" are formed in enterprise. At present, the exchange and cooperation between enterprises increase for obtaining competitive advantage. Therefore, many advanced management techniques such as collaborative commerce, agile manufacture, virtual enterprise, enterprise information portal and knowledge management are appeared, which need all applied systems of the enterprise being based on Browser/Server application mode, integrated, open and face to customer of information system. The enterprise can in time adjust product variety and output, adjust craft production process, optimize production and satisfy market needs with maximum limit according to market changing, and achieve modernization integration management of enterprise, only if it carries out the information integration of management decision level, strategic plan optimization level, production scheme optimization level. To solve these problems, two methods are proposed. One method is to give up the existing system and develop a new unify information system after re-designing and re-programming. Although this method is very ideal, this isn't an actually viable solution in consideration of development cost, the implement factors such as period and difficulty. The other method is Enterprise Application Integration (EAI), which is widely used in IT last years.

EAI can unite operation process, application software, hardware and various standards, implement the seamless integration between the two or more enterprise application systems, and make them like a whole carrying on the operation processing and information share [1,2]. In view of the need of enterprise, enterprise application integration platform should satisfy high adaptability, emphasize the management of "with user center" and have the ability of integrating the different information resource. However the method of using the existing simple data integration, information integration and process integration cannot satisfy the requirement of enterprise. Agent and multi-agent techniques developing rapidly in the artificial intelligence has become the research hotspot, which is used widely in many fields. Because multi-agent system has the advantage of cooperatively solving problem, describing problem naturally, and using existing software by integration etc., it can provide new solution for the enterprise application integration [3,4]. Therefore, multi-agent technology was introduced to research the enterprise application integration. Moreover, in consideration of the agency properties and ability such as autonomy, interaction, adaptability, learning and cooperation, if traditional object model was used to design and implement the EAI, which would make complications and hard to learn, maintain and reusability problem etc.. Therefore, aspect oriented technology was adopted to implement separation of the concern point of different agency properties and cooperation ability, to avoid code coping and improve reusability, taking the advantage of aspect oriented design and programming.

2 THE EAI FRAMEWORK BASED ON MAS

Multi-agent system (MAS) technology provides a road of convenience for the enterprise application integration that distributes on time and the space, having already caused the concern of the domestic and international researcher. The Stanford university J Y C Pan and J M Tenebaum put forward the EAI framework based on agent and developed a

few EAI prototypes on this foundation [5]. On the foundation of this framework, this paper put forward the EAI framework as shown in Fig.1.

The EAI platform is designed to be composed of four levels: interface level, collaborative level, application level and data level. Different agent was adopted on each level.

Interface level is designed to be composed of User Interface Agent (UA), User Information Agent (IA), Data Type Abstract Agent (TA). UA is agent, which is used to interact directly with user. It can apperceive the action information of user and receive other agents' information. TA provides interface for interface level and collaborative level to convert different data type from interface level into consistent data type, which can be used in UA. The emphasis of interface level design is researching function analyze of user view, business characteristic and working habit of user, confirming the source of knowledge and the method of learning and training, and establishing the environment of Human-Computer interface.

Collaborative level, which reflects the activity of enterprise business, is designed to be composed of Business Agent (BA) and Optimization Agent (OA). It locates between interface level and application level and is used to connect interface level and application level. BA is corresponding to the activity of enterprise business and is responsible for explaining and executing the message from interface level in order to coordinate the mutual conflicts amongst agents of application level. We know enterprise optimization is very important to production enterprise. The enterprise must optimize the purchase of raw material, the sales of products, and the operating condition of installations in order to reduce costs, improve quality of products such that immediate response to market changes could be made, for upgrading economic performance and keeping the superiority of competence. Therefore, Optimization Agent was adopted in collaborative level in order to get global optimum result.

Application level reflects actual enterprise process and the relation of entities. In general, it is implemented by analyzing user requirement and mapping the space of actual problem into the space of computer. Application level is designed to be composed of Application Agent (AA) and Data Assistant Agent (DA). AA, which is implemented by encapsulating existing application and adding agency property, can cooperate with other applications to solve problems. Because early application systems such as inventory control, human resources, and sales automation are designed to run independently, with no interaction between the systems, each application system accesses respective data (including database, data warehouse, knowledge base and document etc.) and these data resource are independently. In order to interact data easily between systems, in this paper, Data Assistant Agent is adopted to convert different data format to same data format The emphasis of application level design is analyzing application model, encapsulating existing systems to application agents and designing data assistant agent.

Data level is storage management level of enterprise information, including document base (semi-structured document and unstructured document such as Word document, E-MAIL, WEB page), database, data warehouse, knowledge base and debate group document.

3 THE STRUCTURE AND IMPLEMENTATION OF AGENT

3.1 THE DEFINITION OF AGENT

An agent is a complex object whose state is viewed as consisting of metal state (e.g., beliefs, capabilities, choices, and commitments) or additional agent property (e.g., autonomy, adaptability, cooperation) [3]. In general, the state of agent is formalization by knowledge and is expressed by metal component such as beliefs, goals and plans. The model of beliefs is the external environment of agent's interaction. The goals are realized by different plans. The behavior of an agent is depended and affected by cooperation of its property. In fact, there are without a unified definition about the basic components of an agent, that is to say, how to describe the state and the behavior of an agent and which properties are the most fundamental. But now, as a whole, an agent is viewed as an autonomy entity which changes self-state and behavior by interactive with external environment. Therefore, The autonomy, interaction, adaptability are viewed as the most fundamental properties and that the learning, mobility and cooperation are viewed as optional properties. Agents used in application integration which are consisted of User Interface Agent (UA), User Information Agent (IA), Business Agent (BA), Optimization Agent (OA), Application Agent (AA) etc as showed in Fig.2. Each agent has special function and properties. According to the function of agent, EAI system can be classified four kinds of agent. (1) Interface Agent: including user interface agent. (2) Application Agent. (3) Business Agent (4) Assistant Agent: including user information agent, data type abstraction agent, optimization agent and data assistant agent etc. Each agent has fundamental properties. At the same time it has different capability and special properties.

3.2 THE IMPLEMENTATION OF AGENT

Agent can be defined using the form of predicative expressions. For example, the optimization agent in collaborative level can be defined as follows:

```
<optimization agents>:=<identifier><object>
        <communication mechanism><cooperation
        mechanism><metal state><acquaintance model>
        <restriction condition>
<local data description>: =< semantic >
         <syntax><structure><method><property>
         <method interface>
 <communication mechanism>: =<XML creation>
          <XML resolution>
  <knowledge base>:=<event action rule>
           <how to acquire help>< report to who >
 <metal state>:=<beliefs><intention><commitment>
           <responsibility>
 < acquaintance model>:=<identifier>
           <method interface><property>
```

The optimization can be implemented by Java language as follows:

```
CLASS Optimization_Agent
{
 IS-A:(Agent)
 INSTANCE-OF:(none)
 ATTRIBUTES:
        Total_Profit_model{
            Goal:"maximize";
            objective_function:
                Maximize_Total_Profit;
            decision_variable{
                name:string; indices:string;
                non-negativity:[yes|no];
                domain:[Real|Integer|Binary]; };
```

CONSTRAINT [1]: Minimum inventory limit;
CONSTRAINT [2]: Maximum inventory limit;
CONSTRAINT [3]:Inventory balance;
CONSTRAINT [4]: Backlog;}
OPERATIONS：
Interpret (input:$M_{encoded}$;output:ModelTemplate);
createXML();
analyseXML(xmlfile : *file);
Instantiate_Model(input:ModelTemplate,DataSe;
output:SourceModel）;
Execute_Model_Insatnce(input:SourceModel;
output:Outcome);
Update_Model(input:SourceModel,UpdateInformatio
n;output:UpdateModel);
Ask (solve ($M_{Mps\_format\_file}$));
Tell (M_{output});
Advertise ($M_{changed}$);
}

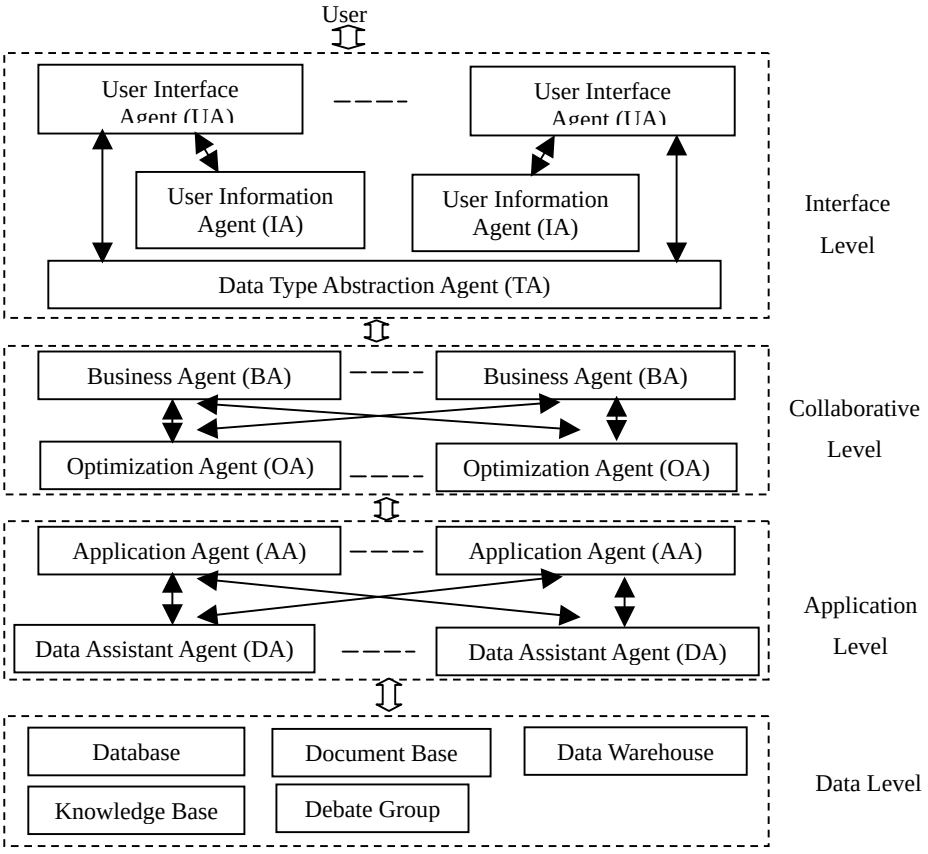


Fig. 1. The Framework of EAI Based on MAS

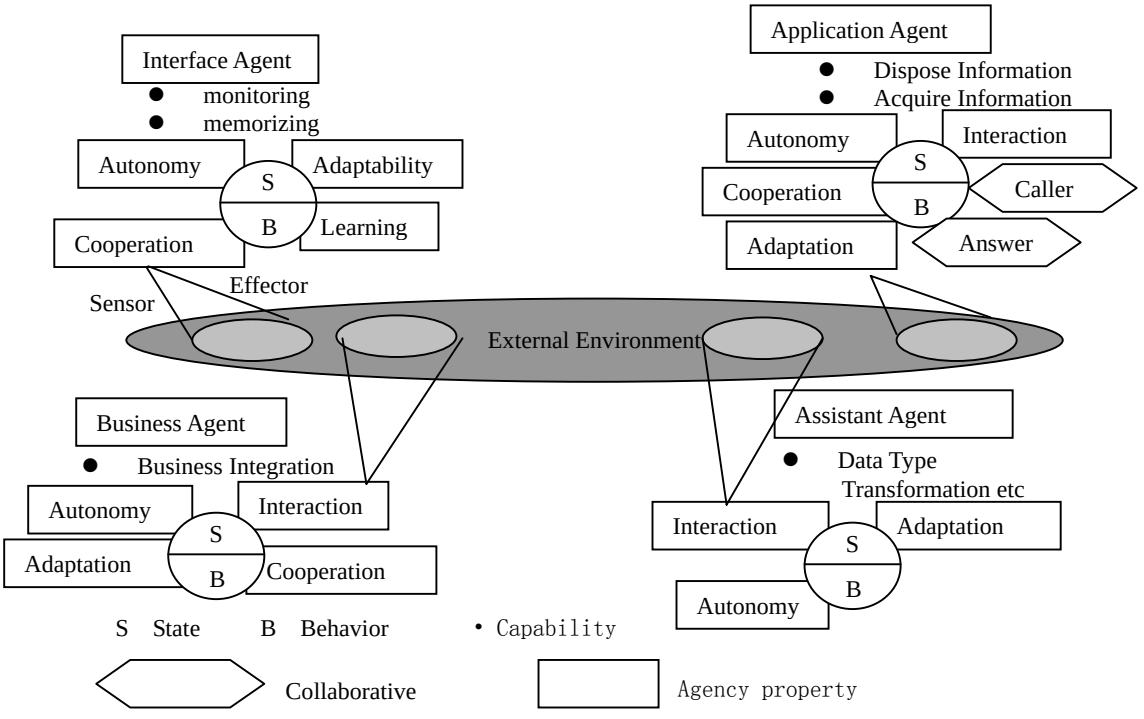


Fig. 2. The Structure of Intelligent Agents in Enterprise Application Integration

4. THE IMPLEMENTATION OF SYSTEM BASED ON ASPECT

The implementation of agent proposed above is the agent viewed as an object, which encapsulates behaviors and states together. But from the point of actual process of development, this method is adverse to design and implementation of agent, and few shortcomings are found as follows:

(1) It is difficult to understand, maintain and reuse because the object is very complex.

(2) It is difficult to design rationally because the developer needs to think over many properties of MAS.

(3) Many common function such as exception check/handling, log, trace and synchronization are needed to design repeatedly and cannot be reused.

Therefore, this paper adopts aspect-oriented technology to construct the model of agent and implement the multi-agent system. It can overcome above shortcomings by using the advantage of design and programming of aspect.

4.1 ASPECT-ORIENTED TECHNOLOGY

Object oriented programming and design have become main stream over the last years, having almost completely replaced the procedural approach. One of the biggest advantages of object orientation is that a software system can be seen as being built of a collection of discrete classes. Each of these classes has a well-defined task and its responsibilities are clearly defined. In an OO application, those classes collaborate to achieve the application's overall goal. However, there are parts of a system that cannot be viewed as being the responsibility of only one class, they cross-cut the complete system and affect parts of many classes [6]. Examples might be locking in a distributed application, exception handling, or logging method calls. Of course, the code that handles these parts can be added to each class separately, but that would violate the principle that each class has well-defined responsibilities. This is where AOP comes into play: AOP defines a new program construct, called an aspect, which is used to capture cross-cutting aspects of a software system in separate program entities. The application classes keep their well-defined responsibilities. Additionally, each aspect captures cross-cutting behavior. This approach has the following advantages [7,8]:

(1) Different concerns are implemented in different aspect modules. As a result, it fully implements the separation of cross-cutting concerns in formality and improves the readability and maintainability of programming code.

(2) It improves reusability by providing a common approach (aspect module) relative to class in OO to express concern.

(3) It implements perfect joint of component and aspect by providing the mechanism of composing concern. Aspect-oriented programming includes the composing factors of general programming. In fact, if aspect-oriented programming didn't use aspect, it would become general programming.

(4) It does not change the mechanism of programming language. Therefore, Aspect-oriented programming language is better to unite general programming language and is facility to developers. For example, AspectJ is facility to embed in the IDE of Borland JBuilder.

(5) In a certain extent, it simplifies the development process of system. Traditional development process is complex because of there are anfractuous relations between each other modules. But, the aspect approach reduces the complexity of software development process by resolving crosscuts between modules.

The implementation of agent, which is based on traditional object model, was discussed in Section 3.2. In order to solve existing problems about object model, in the following, our agent design model is presented as an aspect-oriented extension of the traditional object model.

1) Agent's Core State and Behavior. The Agent class specifies the core state and behavior of an agent, and should be instantiated in order to create the application's agents. Since an agent is described in terms of its goals, beliefs, and plans, the attributes of an Agent object should hold references to objects that represent these elements, namely belief, goal and plan objects [9]. Methods of the Agent class are used to update these attributes and implement agent's capabilities.

2) Agent Types. Our EAI system proposes the use of inheritance in order to create different agent types. Different types of agents are organized hierarchically as subclasses that derive from the root agent class. The methods of these subclasses implement the capabilities of each agent type. Fig.3 illustrates the subclasses representing the different kinds of agents of our case study (Section 3.2).

3) Agency Aspects for Agenthood. In order to improve reusability, maintainability and readability, each agency aspect is responsible for providing the appropriate behavior for an agent's property. An agency aspect introduces an interface related to the agent's property, and implements the advices that crosscut the core agent's functionality. Fig.3 depicts the agent aspects, which define essential agency properties for agenthood: interaction, adaptation, and autonomy. For example, when the interaction aspect is associated with the agent class, it makes any agent instance interactive. In other words, the interaction aspect extends the Agent class's behavior to send and receive messages.

This aspect updates messages and senses changes in the environment by means of sensors and effectors. Since the process of sending and receiving messages occurs quite often in multi-agent systems and cuts across the agent's basic capabilities, the implementation of this process as an aspect is a design decision that avoids code duplication and improves reuse. The Autonomy aspect makes an Agent object autonomous, it encapsulates and manages one or more independent threads of control, implements the acceptance or refusal of a capability request and for acting without direct external intervention. The advice MakesDecision() implements the decision-making process by invoking specified decision plans. The advice PerformsPlan() implements the ability to accept or refuse a capability request. The adaptation aspect makes an Agent object adaptive, it adapts an agent's state and behavior according to new environmental conditions. The AdaptBeliefs(), AdaptGoal(), AdaptPlan() advices are responsible for updating beliefs, goals, and plans,

respectively.

4) Particular Agency Aspects. The agency aspects that are specific to each agent type are associated with the corresponding subclasses. Note that the different types of software agents inherit the agency aspects attached to the Agent superclass. As a consequence, the user interface agent, application agent, business agent etc. inherit agenthood features and only define their specific capabilities and aspects.

5. CONCLUSIONS

A larger chlor-alkli chemical enterprise which mainly produces PVC (polyvinyl chloride) was taken as application example in our work. It has a number of application systems such as production management system, inventory management system, financial management system, and human resources. But existing systems are generally applied in special function section and manage different object, and each system runs independently and lacks effective communication because of history. As a result, "the isolated information islands" are formed in enterprise. It adopted EAI in order to improve the management level and the competition ability. In the process of EAI, multi-agent framework of EAI was adopted to solve the problems of application integration. Otherwise, aspect-oriented technology, which has the advantage of aspect, oriented design and programming was adopted to implement each property of agent and separation of crosscutting concerns. As a result, it can avoid coding copy and improve reuse ability. The method proposed in this paper has obtained a better result according to the process of development and running.

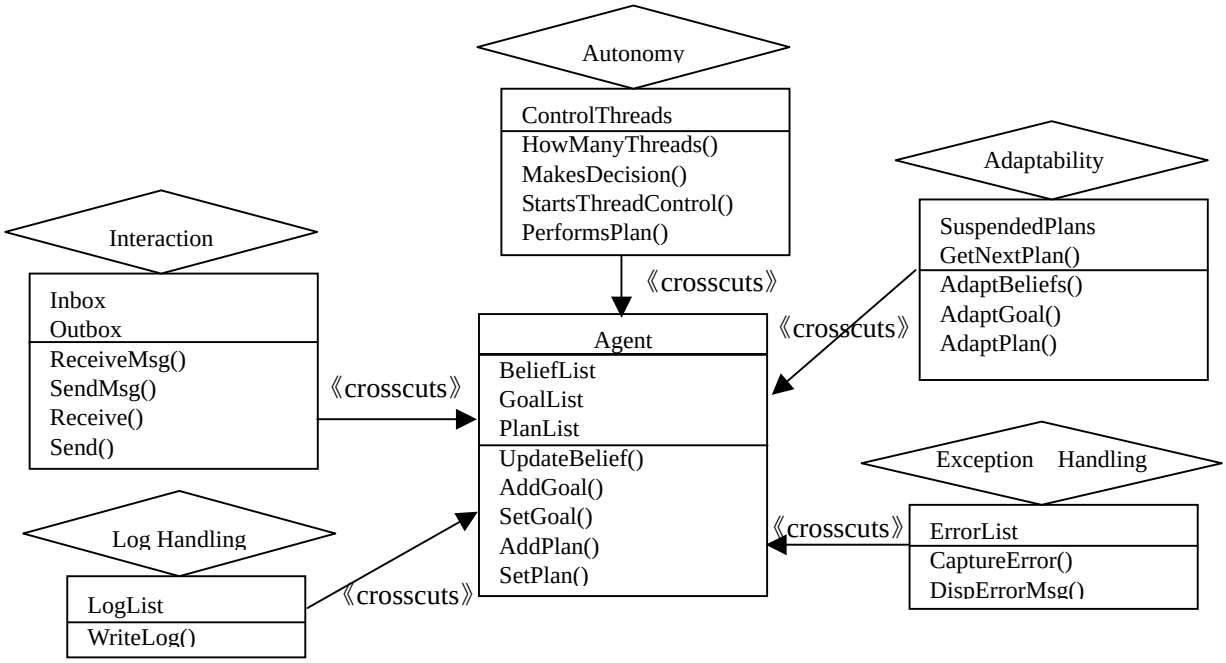


Fig. 3. The Aspect Structure of Intelligent Agent

6.REFERENCES

[1] S. L. David, Enterprise Application Integration [EBOL], http://eai.ebizq.net/.

[2] L. Yeamans, Enterprise Application Integration, NSAG, Inc., 1999, http://wwww.messageq.com/EAI_survival.html.

[3] Y. Soham, "Agent-oriented programming", Artificial Intelligence, 1993(60): 51-92.

[4] M. Wooldrige, N. R. Jennings, "Intelligent agents: Theory and Practice", The Knowledge Engineering Review, 1995,10(2): 115-152.

[5] J. Y. C. Pan, J. M. Tenenbaum, "An Intelligent Agent Framework for Enterprise Integration", IEEE Transaction on Systems, Man and Cybernetics, 1991, 21(6): 1391-1407.

[6] M. Aksit, B. Tekinerdogan, et al "The Six Concerns for Separation of Concerns", Proceedings of ECOOP 2001 Workshop on Advanced Separation of Concerns, Budapest, Hungary, Jun.2001, 18-22.

[7] T. Elrad, R. E. Filman, et al "Special Issue on Aspect Oriented Programming", CACM, Vol.44, Oct.2001.

[8] T. Elrad, R. E. Filman, et al, "Aspect-Oriented Programming", CACM, Vol.44, Oct.2001, 29-30.

[9] F. G. Alessandro, J. P. Carlos, "An Aspect-Based Object-Oriented Model for Multi-Agent Systems", ECOOP 2000 workshop on Aspects and Dimensions of Concerns, 2001.

Optimize adaptive self-organization mechanism for wireless sensor networks

Wu Bi-Yan
College of Computer Science & Information Engineering, Zhejiang Gongshang University
Hangzhou, Zhejiang Province, P.R.China, 310035
Email: kelly@mail.zjgsu.edu.cn

ABSTRACT

Aimed at manage numerous wireless sensor nodes, some mechanism of adaptive self-organization mechanism for wireless sensor networks (WSN) is submitted, however not efficient. Optimizing adaptive self-organization mechanism for WSN, one proposed is optimizing partition nodes in different scale sensor network. The performance of difference partition nodes is demonstrated by simulation in this paper. We analyze The WSN average cover lossy ratio, average number of covered nodes with variant hop count and average delay under different environment include the Minimum hop count and free hop count, Density effect, the classical 20/80 rule with different HMNR, Various number of nodes with fixed the number of header and Various number of nodes with fixed the HMNR is 20/80.

Keywords: wireless, sensor, networks, node, delay, cover lossy ratio

1. mechanism overview

Recently MEMS, wireless communications, digital electronics devices have been developed as cheap, portable, low power, tiny and multifunctional sensor nodes. Sensor nodes may be spread over a large area and long distances and multi-hop communication may be required between nodes. Emerging wireless sensor network (WSN) comprises various nodes that combine the capabilities of conventional sensors with those of low power processors and wireless communication modules [1-2]. Aimed at manage numerous wireless sensor nodes, some model and mechanism of adaptive self-organization mechanism for WSN is submitted, however not efficient [3-4]. Optimizing adaptive self-organization mechanism for WSN, one proposed mechanism reduces the number of management nodes in large-scale sensor network, so we can reduce the cost of managing nodes and of the communication among them [3-5]. The nodes of lower level are controlled and organized by the nodes in the higher level. So the right ratio of headers to member nodes can improve performance of WSN obviously. The Simulation and analyze effect of the ratio of headers nodes is discussed in this paper on TinyOS environment based on Mote.

The performance of difference partition nodes is demonstrated by simulation in this paper. We analyze The WSN average cover lossy ratio, average number of covered nodes with variant hop count and average delay under different environment include the Minimum hop count and free hop count, Density effect, the classical 20/80 rule with different HMNR, Various number of nodes with fixed the number of header and Various number of nodes with fixed the HMNR is 20/80.

2. simulation environment and method

We demonstrate the performance of our approach by simulation. The simulation environment is TinyOS. The TinyOS is assumed to run on a UCB mote [5] that uses the power characteristic to model energy consumption. MICA series Motes were originally developed at University of California at Berkeley [6] and manufactured by Crossbow Technology Inc. [7]. We refer to this platform and use TinyOS simulator (TOSSIM). The proposed algorithm is implemented in TinyOS with various numbers of nodes, which are randomly deployed over 100*100 to 300*300 square units. The number of sensors is varied from 80 to 160, and the number of headers is fixed at ten, the Header-to-Member node ratios (HMNR) are 4:6, 3:7, 2:8 and 1:9, which is shown at Table 1.

Table 1 Number of headers

Nodes	Number of headers			
	4:6	3:7	2:8	1:9
80	32	24	16	8
100	40	30	20	10
120	48	36	24	12
140	56	42	28	14
160	64	48	32	16

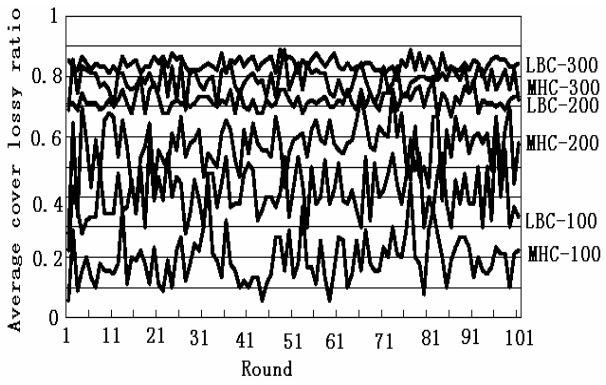
The channel is assumed to allow collision and packets may be dropped in the medium. Link error probabilities are proportional to the distance between nodes. The asymmetry of the links between any two nodes is modeled. TOSSIM provides a radio abstraction of directed independent bit errors between two nodes. The independence of the bit error means that longer packets have a high probability of corruption and the loss probabilities of all packers are mutually independent [8].

The environment is an area of 100*100~300*300 square units. 100 nodes that include ten header nodes are deployed randomly. Using CSMA base and wireless link are not perfect, but with error link probability. The approach involves a low cover loss in all cases. Multi-hops are used to cover the member nodes. Some results obtained by simulation the proposed method. The performance metrics as followed is elucidated: average cover lossy ratio (ACLR), Range of cover loss (RCL), average number of covered nodes with variant hop count and average delay. The Average cover lossy ratio is abtained through measurement total number of nodes that are not covered after clustering. Average delay is the total average delay from member nodes to headers in all clusters.

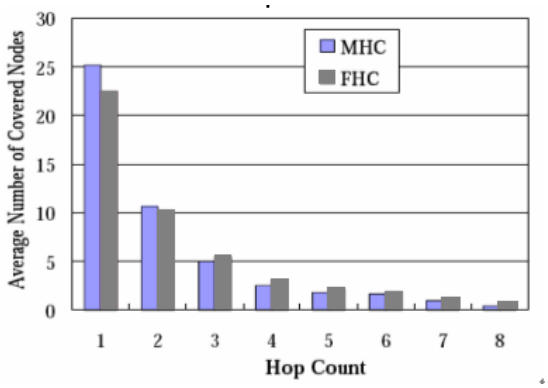
3. Performance and results analysis

3.1 compare the Minimum hop count (MHC) with free hop count (FHC)

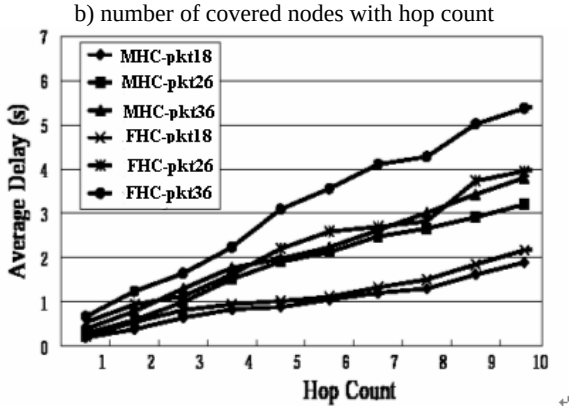
Member nodes randomly generate the event data, and the packet length is varied from 18, 26 and 36 bytes for sending. The FHC is modified from MHC, and covers number nodes by the free hop count. However, the hop count of MHC is less than that of FHC and FHC has a longer delay because the longer hop nodes compete with the short hop node for use the same path to reply to the header. The result is presented in Figure 1.



a) aaverage cover lossy ratio with 100 cover round



b) number of covered nodes with hop count



c) Average delay with hop count

Fig.1 comparison between MHC and FHC

Average cover lossy ratio increase with the increasing of the cover number nodes. And the Average cover lossy ratio of MHC is lower than that of FHC with the same cover number nodes (Figure 1a). With the increasing of hop count, The number of covered nodes decrease. Number of covered nodes of the MHC is larger than that of the FHC with the same hop count (Figure 1b). Average delay increase with the increasing of hop count, however average delay of MHC is less than that of FHC (Figure 1c). Those change curves accord with the adaptive self-organization mechanism for wireless sensor networks, however MHC show a better performance than FHC.

3.2 Density effect

We zoom in/out from an area of 100*100 to one of 300*300 is performed, while maintaining the relative location of the nodes. One hundred nodes are randomly deployed and include ten headers. When we zoom in/out the area will make different density. A higher density setting corresponds to low cover loss. The TinyOS sets the link probability according to the distance. Varied from an area of 100*100 to one of 300*300, the the link probability is changed accordingly. A short distance between two nodes corresponds to a strong link. With the increasing of net area and average cover lossy ratio (ACLR) and range of cover loss (RCL) increase owing to the increasing of net nodes. Therefore, a high-density setting yields a low cover loss. The result is listed on Table 2.

Table 2 Various cover lossy ratio with variant area

area	100* 100	150* 150	200* 200	250* 250	300* 300
ACLR	0.1	0.23	0.41	0.61	0.73
RCL	0.2	0.3	0.24	0.16	0.10

We can observe the sparse case more stable. In 300*300 case the range of cover loss is 0.1. The wireless link is weak in sparse case. A parent node can not forward CREQ to children nodes, it goes without saying the children nodes can reply cover respond (CREP) to parent node. The wireless link disconnecting is certainly. But other case children nodes may receive the the cover request (CREQ) from parent node, but whether can send back to header depend on reply link probability. This results remind us setting the asymmetric of wireless link.

3.3 The classical 20/80 rule with different HMNR

The cover loss with various proportions of headers is listed on Table 3. One hundred nodes are deployed randomly in 200*200 square units and HMNR are set from 1:9 to 4:6. From the table 3, we can see that different header-to-member node ratios (HMNR) accord with different cover lossy ratio, the more headers yields lower cover loss, the more children nodes yields high net area and average cover lossy ratio (ACLR) and range of cover loss (RCL). The high HMNR, which is means the more headers, will converge quickly. The nodes with fewer hops will have a high probability and can reply by sending CREP to header successfully. Therefore, a appropriate HMNR value that yields low cover lossy ratioc and show a good performance can be obtained. But there exits no a according value of HMNR for both lowest ACLR and RCL.

Table3 various cover lossy ratio with variant HMNR

HMNR	1:9	2:8	3:7	4:6
ACLR	0.37	0.29	0.28	0.26
RCL	0.24	0.23	0.17	0.21

3.4 Various numbers of nodes with fixed the number of header

Figure 2 present the cover lossy ratio with various nodes. We randomly deployed number of nodes from 80 to 160 and fixed the ten headers in 200*200 square unites area respectively.

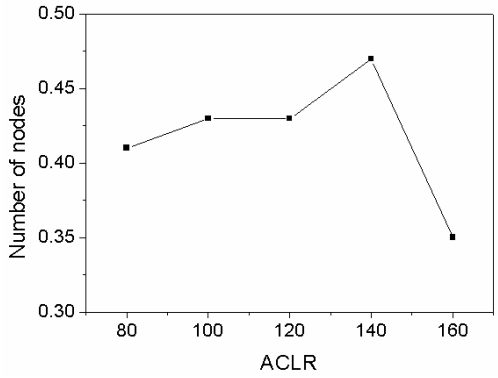


Fig. 2 Various cover lossy ratio with number of nodes

With the increasing of number of nodes with fixed the number of header, the ACLR firstly increase and then decrease. There is a extremum according to a optimal number of nodes. Various cover lossy ratio with When we increase the number of nodes will make the higher density but lower HMNR relatively. The two factors will counteract each other for reduce cover loss. But we can see the density can against the cover loss finally, and the maximum range of average cover loss is 0.13. The number of nodes distribute in each hop count is increase with high density.

3.5 Various number of nodes with fixed the HMNR is 20/80

The number of nodes is changed from 80 to 160, but the same HMNR, is maintained in the 200*200 square units. In all case, the cover loss is as shown in Figure 3.

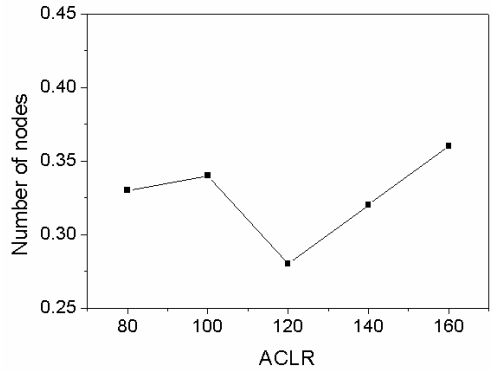


Fig. 3 Various cover lossy ratio with number of nodes

With the increasing of number of nodes with fixed the number of header, the ACLR firstly decrease and then increase. There is a extremum according to a optimal number of nodes. The HMNR which is 2/8 can obtain a lowest ACLR and best cover loss. 20/80 rule is always applied in many field needed optimal ratio. Hence the 20/80 rule is good guide for leading the HMNR and can be useful applied to wireless sensor networks. The maximum range of average cover lossy ratio is 0.07. The average number of covered nodes with hop count shows the same behavior of converge in different case.

Both of the node density and the number of headers can be varied to improve cover loss. For a given density, changing the number of headers alters the cover loss. The number of headers is determined by the 20/80 rule, which is a good guide for wireless sensor networks. The 20/80 rule, as applied to leadership in a hierarchy sensor network may actually be from 40/60 through 10/90. However, 20/80 is experimentally supported, and the performance is approximate for 30/70 and 40/60.

4. Conclusions

Through analyze The WSN average cover lossy ratio, average number of covered nodes with variant hop count and average delay under different environment include the Minimum hop count and free hop count, Density effect, the classical 20/80 rule with different HMNR, Various number of nodes with fixed the number of header and Various number of nodes with fixed the HMNR is 20/80. We think that the 20/80 rule is experimentally supported and the performance is approximate for 30/70 and 40/60. The proposed algorithm guarantees the approximate cover loss and scale of large networks, in adjusting the number of nodes and maintaining the ratio of headers to nodes. A moderately high density will strengthen then links and yield a low cover loss.

Acknowledgements

The authors wish to acknowledge Doctor Jiang Xiaobo in Micro-electronic Institute of China Science Academy for his help in running the program.

Reference

[1]Akyildiz L.F., Su W.L. A Survey on sensor networks. IEEE Communications Magazine, Vol. 40.No.4, 2002, pp.102-114

[2]Yu Y., Estrin D., Govindan R. Region Based Multipoint Data Dissemination in Wireless Sensor Networks. www.lecs.cs.ulcs.edu/~yanyu/pub/papers /gear_Journal_in_sub.pdf,2004,12,16

[3]Kim S., Sony S.H., Stankvicy J.A., A data dissemination protocol for periodic updates in sensor networks. www.cs.virgnia.edu/~control/docs/papers, 2005,2,26

[4]G .Gupta, M.Younis, Load-balanced clustering of wireless sensor networks. Proceedings of the international conference on communication. Vol. 13 No. 5, 2003, pp.1848-1853

[5]J. Hill, R. Szewczyk, A.Woo, System architecture directions for networked sensors. Proceedings of ACM SIGPLAN. Vol. 35, 2002, pp.93-104

[6]www.tinyos.net/scoop/special/hardware/

[7]Crossbow Technology Inc. (www.xbow.com/)

[8]P.Levis, N.Lee, TOSSIM: a simulator for TinyOS Networks. User's manual. www.cs.berkeley.edu/~pal /pubs/nido.pdf. 2003,6

An Optimized Algorithm of Node Selection for Wireless Sensor Network

Xianfang Wang
Henan Institute of Science and Technology
Xinxiang, Henan 453003, China
Email: xfwang11@sina.com
Zhiyong Du
Henan Mechanical and Electrical Engineering College
Xinxiang, Henan 453002, China
Email: zhydu@163.com

ABSTRACT

At first, this paper introduced the features and applications about wireless sensor network briefly, and put forward that the selection of the network nodes and the optimization of wireless sensor networks are very important aspects. Then it researched the problems about the selection of wireless sensor network nodes. Based on the principles of the algorithm performance analysis, it pointed out some optimizing strategies. Through relevant simulation experiments, it proved that the optimized algorithm is more accurate and efficiency, it needs to be popularized and applied.

Keywords: Wireless Sensor Network, GB Algorithm, Node Selection, Coverage range, Cluster.

1. INTRODUCTION

In recent years, with the rapid development of sensor technology, computer networks, microelectronic mechanical systems and signal processing technology, wireless sensor networks gradually had become a hot topic.

Wireless sensor networks will be exposed to a large number of sensor nodes deployed or interested regional, the node forms a wireless network quickly through its own organization. Each node has its own controlling region, through the perception of equipment, such as temperature, humidity, sound or optical equipment, chemical analysis devices, electromagnetic induction device, to monitor the surrounding physical environment. It also can realize the specific environment functions through configuring some special functional models.

Generally, the communication distance of the nodes is short, it can only exchange data with their only communication nodes within other sensors (or neighbor called neighbor nodes). If communication nodes are visited outside, it must use multi-hop routing. Therefore, the node is not only the collector of information but also the transmitter. The gathered data get to the gathering nodes by many routs (also called the gateway some literature). Gathering node is a special node, it can communicate with control center through the Internet, mobile communication networks, satellite [1,2,3]. It can also gather gateway data through letting the aircraft fly over the network. To ensure that the majority of nodes to connect wireless links, sensor nodes have good coverage on the target region, the nodes distribution must be suitable crowded. However, the increase in the numbers of nodes that will increase the difficulty of data collection and processing, thereby affecting the speed of operation of the network, the selection and optimization of the network nodes are wireless sensor networks a very important aspect.

2. SELECTION OF NETWORK NODE

2.1 General methods of node choice

Choice of sensor network nodes is determined by implement communications, processing or communications task nodes, and choose to enter the dormant state of the entire network of nodes to extend the lifetime.

The simplest method is to select all nodes in network to collect environmental data, this approach almost has no additional cost. But due to the property of dense distribution of sensor network nodes, the info-redundant of the gathering node is the largest. It leads to low efficiency and high energy consuming.

To avoid all nodes to collect data, it can be randomly selected parts of nodes to do this work. Random choice algorithm is a simple way, small additional cost, but still has some redundant nodes choice, or unable to meet user demands for information, and it must to add certain feedback control mechanisms.

Yan and other people pointed out the methods of choosing nodes [4,5,6]. Its basic idea is that each sensor node gathers some useful data in its coverage area of collecting, the coverage of all nodes combination form the systemic coverage. But nodes randomly spread may lead to cover redundancy. Therefore, it is necessary to determine the minimum subset, it covers all the regions, and any node out of this subset is in inactive situation. In this way, it will not only save energy and reduce potential conflicts in wireless channels, reducing the possibility of the use of visiting media. Thus it extended the lifetime of network, and enhanced network performance. But, the calculation of this method is large, and the additional telecommunication costs too.

2.2 GB (Grid-Based) node choice algorithm

Chen followed the idea of coverage, he presented choosing methods based on GB [7]. Its basic idea is to create a grid, the use of grid cross-point (known as sampling points) to describe the sensor nodes cover region, was chosen for the assembly nodes of the original network coverage of all the sampling points. Here, it is assumed that the sensor nodes have no mass movement during the period of implementation algorithms, and each node (node i) know their own coordinates.

Algorithms procedures are the following:

1) Coverage sampling stage. On the basis of node coordinates and grid density d could receive the assembly P_i of sampling points by node i covered, which is its own coverage. And it received the total cover area that was covered with M-node network P, $P = \bigcup_{i=1}^{M} P_i$.

2) Pretreatment phase. Each sampling point might be covered with many nodes, it presume that the assembly of nodes covered with a same sampling point n is Q_n, then Q is the assembly of nodes covered with all sampling point N, it is clear that $Q = \bigcup_{i=1}^{M} Q_n$. If $Q_m = Q_n$, these sampling point m and n could be merged, thus creating a new sampling points assembly P_{new} .

3) The assembly selection phase. Determine the smallest K, it makes the assembly of sampling points covered with K-node equal to P_{new}. This is a NP problem, only through greedy algorithm or linear planning models explained and resolved.

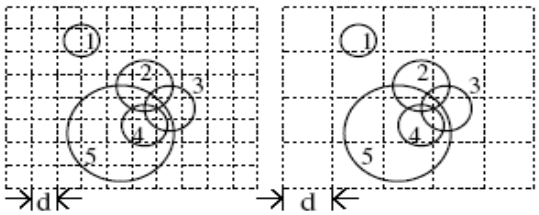
2.3 The realization of distributed algorithm

If use a central node to carry out the GB node choice algorithm, it would cause the performance bottleneck and the heavy signal expenses. Chen proposed one kind of node choice algorithm distributional realization to reduce the signal expenses and disperse calculated loading. After the cluster formed, it carried out the GB algorithm by take the cluster as the unit. Each node transmits a report new to cluster-head, including cluster-class information, such as node ID, cluster-head ID, routing from node to cluster-head, and so on; also including cover information, such as coordinates, node cover scope, etc. It is a supposition that the report information by the member of cluster transmitted all can arrive at a cluster-head node successfully, once the node is chose for the working node, cluster-head will transmits some information to this node. Which node has not received the news, it would be inactive. Very obviously, the distributional solution reduced the system expenses with taking the result precision as the price.

3. THE PERFORMANCE ANALYSIS AND OPTIMIZATION OF GB ALGORITHM

3.1 Performance analysis

1) Grid density d can affect the accuracy of algorithms and timing complex degrees. As shows in Fig. 1, Fig. 1 (b) is a sparsely grid, the node of covering different regions might contain the same sampling points assembly, and regional coverage of the node do not contain any sampling point may be overlooked, which may lead to inaccurate coverage sampling. If increased the density of grid, as shows in Fig. 1 (a), while improving the accuracy, but increased algorithm complexity. Therefore, in the pre-period of the choice node, it is first necessary to find a suitable d in meet accuracy requirements while maintaining a lower degree of complexity.



(a) (b)

Fig. 1. The sample of different density grid

2) The node density aspects the number of selected working nodes. Experiments have shown that when node density larger, the number of nodes by linear planning algorithm choice is 25% less than greedy algorithm, the cost is calculated using longer time. In large-scale sensor networks, the time spent large complex, linear planning even may be ineffective. It is discovered that when the node density would reach a value after the option is no longer working to increase the number of nodes, the nodes have been selected because of the almost total coverage of the entire region.

0n the assumption that has 40 nodes in the study system, a cluster of the Max-Min D-cluster algorithms Amis formation of a cluster categories [8], designated radius is 10. Fig. 2 shows the number of selection working nodes in the different node density. This shows when node density is small, the total number of nodes rarely, the overlap of covering regional nodes can be ignored, the choice nodes is not very necessary, because almost all the nodes must be chosen, in this case, it will choose more nodes to work; When the larger node density, the more number of total nodes, the overlap part by several regional nodes covered must be considered, it can only selected one node from many nodes of covering the same sampling point for the working nodes, the others in a non-active state.

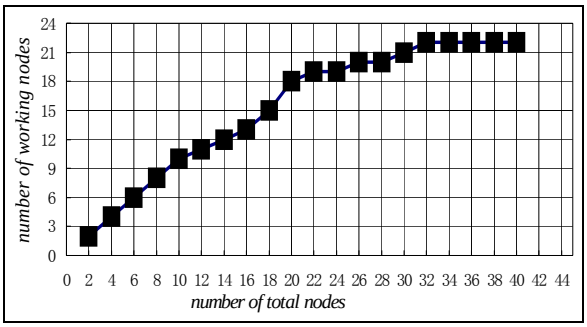


Fig. 2. The number of selected working nodes in different node density

3) The impact of the time complexity about node density. Fig. 3 shows the number of clusters in different node density. Obviously, when the network have 10 nodes, formed 8 clusters; when a 20-node-network, 6 formed; when a 40-node -network, formed 2 clusters. The number of clusters is downward trend with the node density increasing.

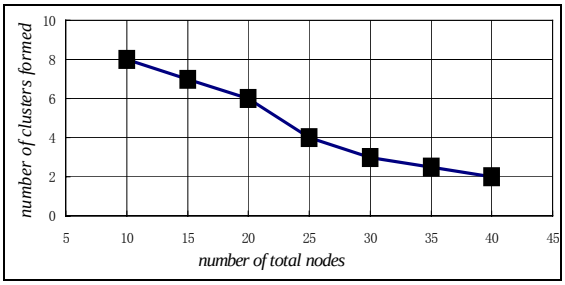


Fig. 3. The number of clusters in different node density

So as to reduce the total number of nodes, the network will produce more clusters, and this will reduce the calculate time to choose the working node.

4) The radius of a cluster-class impact net costs up. Signal spending is defined as the news quantity about the report news and information transmitted between cluster-head nodes and inner-cluster. When a radius of cluster has increased, reducing the number of creatures, the work of choosing the number of nodes is reduced. This is because, when a smaller radius of the cluster, the sensor nodes with two overlapping coverage regional may belong to different clusters; when radius increases, the two nodes may be classified in the same cluster,

so that only one node will be chosen to work nodes. Therefore, the signal spending is increased with the increasing radius of the cluster category. According to the accuracy of demand, it is need to choose a suitable radius size for meeting the high accuracy, and limiting the signals costs at the same time.

3.2 The optimization of the GB algorithms

1) "Replacement" strategy. GB selection algorithm does not consider node failures. If the working node couldn't to work continuously, because of the node's energy was depleted or failure, this might be lead to some sampling points missing the coverage, and affect the accuracy of demand. Therefore, when the energy nodes below the minimum threshold, the node would send a "help" information to a cluster head, tell cluster-head node to find a "replacement" node as the working node, the cluster head sent broadcasting information to inner clusters, search and rescue nodes cover the same sampling points (or substantially the same) node as a "replacement" nodes.
2)The first initiative come enquiries. In the realization of distributed algorithms, it is assumption that the report information by the cluster members transmitted all can arrive to the cluster head successfully, then the cluster head can implement choice algorithm according to the relevant information. However, in the sensor network which distribute a large number of nodes, the phenomenon of lost information is very possible took places, because of many factors, such as routing, congestion and so on. In this issue, the cluster head will take the initiative gather information mechanism. If waiting for a set time, the node has not received the report of the separation of information yet, a proactive mechanism will be start, it makes the node of lost information reissued its news. The mechanism is to ensure that the cluster head can collect the relevant information of all the inner cluster nodes, implement selection algorithms more accurately.
3) Node energy transmission. In the distributional algorithm, each sensor node carries out the distributional algorithm to form the cluster, once the cluster formed, each node transmits a report news to cluster head, this include two kind of information: Cluster class of information and cover information. In this case, if the cover information of two nodes is the same in the cluster (this is very possibly occurs), then chose one node as the working node, these nodes maybe not the most superior, because it is very possible to choose the energy low node to carry out the task. Therefore, when one node transmits the report information to the cluster head, three kinds of information are included: Cluster head information, the cover information and the energy information. In the same cover information, choice highly energy nodes as the working nodes, it will lengthen the lifetime of the network.

4. EXPERIMENT RESULT AND ANALYSIS OF THE OPTIMIZED GB ALGORITHMS

In the same conditions as 3, it also assumed that the system has 40 nodes, the radius of the bluster class is 10, the results of two algorithms comparative as show in fig.4, this is obviously that the number of nodes with optimized algorithm is less than before, this can save the time of data processing, thereby enhancing the network operational speed.

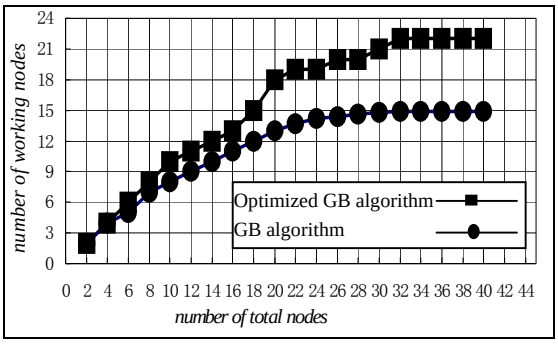


Fig. 4. The comparison of the number of working nodes before and after the optimization under the same condition

5. CONCLUSIONS

In the whole mission distributing system of sensor network, the selection of nodes is very important aspect, because the rationality of selection of nodes can influence the lifetime of network. This paper improved the selecting nodes method of GB. It increases the mechanism of cluster active inquire, replacement strategy and the transmission of node energy. It proved by the experiments that the improved method is more accurate and efficient.

6. REFERENCES

[1] Y.J.Xu, L.Y.Liu, et al, Low Power Processor Design for Wireless Sensor Network Applications, International Conference on Wireless Communications, Networking and Mobile Computing (WCNM 2005), Wuhan China, September 23-26, 2005, pp921-924.

[2] Yannis Kotidis, Snapshot Queries: Towards Data-Centric Sensor Networks, In Proceedings of the 21st International Conference on Data Engineering (ICDE), Tokyo, Japan, April 5-8, 2005, pp 131-142.

[3] R. Szewczyk, J. Polastre, et al, Lessons From A Sensor Network Expedition, In Proceedings of the 1st European Workshop on Wireless Sensor Networks (EWSN), Berlin, Germany, January 19-21, 2004, pp 307-322.

[4] J. Byers, G. Nasser, Utility-based decision-making in wireless sensor networks, *MobiHOC 2000*, Aug. 2000, Pages: 143-144.

[5] D. Tian, N. Georganas, A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks. Wireless Communications and Mobile Computing Journal, 3(2): 271-290, March 2003.

[6] T.Yan, T.He, et al, Differentiated Surveillance for Sensor Networks, In Proceedings of Sensys 2003.

[7] H.Chen, H.Wu, et al, Grid-based Approach for Working Node Selection in Wireless Sensor Networks, IEEE International Conference on Communications, June 2004, Pages 3673-3678.

[8] A. D. Amis, R. Prakash, Max-min d-cluster formation in wireless ad hoc networks, Proc of IEEE INFOCOM 2000. Tel-Aviv, Israel: IEEE Computer Society Press, 2000. 32～41.

A Dynamic Temporary Agent in Hierarchical Mobile MPLS*

Xingchuan Yuan
Wuhan Digital Engineering Institute
Wuhan, Hubei Province, 430074, China
Qingping Wang
Physics & Electronic Department, Wuhan Institute Of Education
Wuhan, Hubei Province, 430205, China
Qingchun Yu
Computer School, Wuhan University
Wuhan, Hubei Province, 430079, China
Lishan Kang
The State Key Lab Of Software Engineering, Wuhan University
Wuhan, Hubei Province, 430072, China
Yuping Chen
The State Key Lab Of Software Engineering, Wuhan University
Wuhan, Hubei Province, 430072, China

ABSTRACT

Reformative Hierarchical MPLS (RH-MPLS) and Adaptive Hierarchical Mobile MPLS Scheme (AHMMS) were proposed. According to RH-MPLS and AHMMS, a Dynamic Temporary Agent In Hierarchical Mobile MPLS (DTAHMM) was proposed. In AHMMS the key problem is the selection of LER in a H-MPLS domain. As the LSP in a domain usually includes at least two LER. Which LER is the most suitable is necessary to be discussed. So the paper proposed DTAHMM. The performance analysis shows that DTAHMM is somehow better than AHMMS.

Keywords: H-MPLS; AHMMS; DTAHMM.

1. INTRODUCTION

The forwarding of MPLS [1][2] is a new packet-forwarding scheme. Packets are assigned labels at the ingress label switching router (LSR)[1][2] of an MPLS domain. Conventional MPLS does not support mobility. Currently there are proposals to incorporate the Mobile IP [3] protocol with MPLS to support mobility in MPLS networks [4]. In [5], the reference proposes the Hierarchical Mobile IPv6. With the integration of mobile MPLS and Hierarchical Mobile IPv6, Hierarchical Mobile MPLS is proposed [6] (H-MPLS).

The idea behind the H-MPLS is to handle the movements of the mobile node locally. When MN moves within the same MPLS domain, no location update messages are sent to the remote home agent. The position of the MN can be determined locally. Thus, in H-MPLS, the movement of a MN within a domain is transparent to the home agent of MN. This is achieved by introducing a new component, Foreign Domain FDA (FDA)[6], to each foreign MPLS domain. Thus, each foreign MPLS domain consists of two levels of foreign agents. When the MN first enters a foreign MPLS domain, the location update message is sent to the Home Agent (HA) of MN. However, every subsequent location update message is forwarded to the FDA, which is responsible for steering the traffic to the current Foreign Agent (FA), instead of the HA of MN.

This paper will propose a Dynamic Temporary Agent in Hierarchical Mobile MPLS (DTAHMM). The rest of the paper is organized as the following. Section 2 gives the Reformative Hierarchical MPLS. Section 3 discusses an adaptive hierarchical mobile MPLS scheme. Section 4 gives the dynamic hierarchical temporary agent in hierarchical mobile MPLS.

2. REFORMATIVE HIERARCHICAL MPLS

In [6], the author proposes H-MPLS to solve the handoff delay and throughput in mobile MPLS. When MN moves into a foreign domain, it will choose FDA as the access point to receive packets from HA or CN. CN or HA only send message to FDA, so MN is transparent to CN or HA.

The movement of a MN moving within a domain is transparent to the home agent of the MN. This is also achieved by FDA, to each foreign MPLS domain. However, each foreign MPLS domain consists of at least two levels of foreign agents. The level is decided the movement of MN. As is shown in figure 1, FDA is the first level, FA1 is the second level and FA2 is the third level when MN moves into FA2's domain. When MN sends the Registration Message to FA2 and FA2 sends the Registration Message to FA1 instead of FDA, FDA sends the Registration Message to HA. When the HA gets the Registration Message and knows the IP address of the FDA, it will send a label request using LDP protocol to the FDA with the IP address of the FDA as the FEC. The FDA replies with an LDP label mapping message back to the HA and sends a label request to FA1 of the subnet in which MN currently is located. When the label-mapping message arrives at HA, the LSP from HA to the FDA is established. Similarly, FA1 replies with an LDP label-mapping message back to the FDA and when this message arrives at FDA, the LSP from FDA to the FA1 is established. So do FA1 and FA2. Then, the HA will search the label table, change the outgoing port and out label into the same values for the LSP from HA to FDA. The HA sends a registration reply to the FDA along the LSP from HA to FDA and the FDA will forward this registration reply to FA1 then to FA2 along the LSP from the FDA to FA1 and LSP from FA1 to FA2. Figure 2 shows the registration procedure of MN in Reformative Hierarchical Mobile MPLS (RH-MPLS) .

* This paper is supported by Chinese National Nature Science Foundation (NO.60133010, NO.60473081).

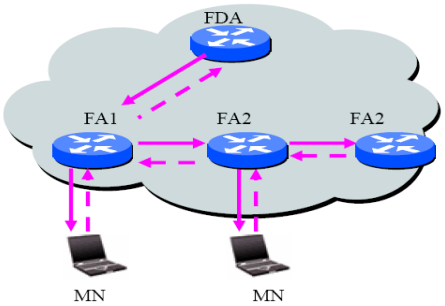


Fig. 1. RH-MPLS architecture

When the MN handovers from one subnet to another within the same foreign domain, it will send a registration request to the new FA. Then the new FA (i.e. the FA2 in Figure 1) will forward the Registration Message to the old FA(i.e. the FA1 in Figure 1), which will send a label request message back to the new FA. The new FA receives the label request and responds back to the old FA with a label-mapping message. Finally, a new LSP, which will be added to the LSP from HA to old FA, will be established from the old FA to the new FA.

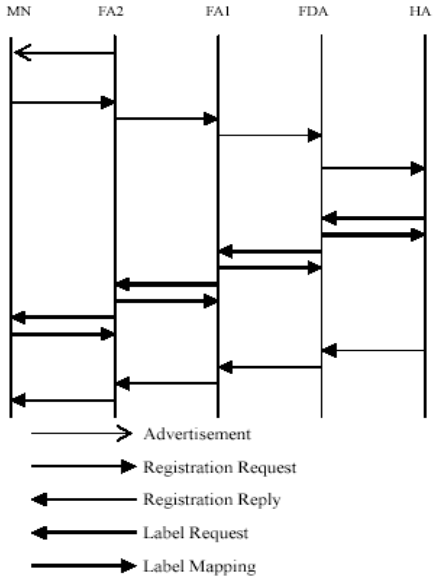


Fig. 2. Mobile node registration in RH-MPLS

3. AN ADAPTIVE HIERARCHICAL MOBILE MPLS SCHEME

Sometimes it doesn't perform better than H-MPLS. So the author wants to integrate RH-MPLS and H-MPLS. In this section, the paper will discuss such integration.

As can be seen in Figure 3 and Figure 4. The key problem is to decide the ingression of LSP when comparing the architecture between RH-MPLS and H-MPLS. FDA or FA1. If FDA is better, we choose FDA. Otherwise, we use FA1. If MN continuously enter into foreign domain, the number of new LSP like that between FA1 and FA2 will increase. So the complexity of administration will also increase. The paper believe that, when MN is not receiving packets, MN should send Registration Message to FA, then FA send the message to FDA. The previous established LSP should be released. The paper also believes that, when MN is receiving packets, MN should send Registration Message to new FA and FDA, if the handoff delay of new FA is smaller than FDA, then new FA send the Registration Message to old FA. Otherwise, new FA send it to FDA.

The Adaptive Hierarchical Mobile MPLS Scheme (AHMMS) integrate the RH-MPLS and H-MPLS. The integration takes the advantage of RH-MPLS and H-MPLS. Maybe (but not absolutely) the end-to-end delay from FDA to MN will increase because the distance is longer, however, the handoff delay will decrease. In mobile MPLS, the object is to decrease the delay.

4. THE DYNAMIC HIERARCHICAL TEMPORARY AGENT IN HIERARCHICAL MOBILE MPLS

In some cases, a LSP from FDA to FA will across more than two LER. Among these LERs, there must be a LER that spending the least time on sending message to MN.

According to AHMMS, MN only chooses FDA or old FA as the temporary agent. Besides FDA and FA, we can also use other FA as the temporary agent, which can decrease the end-to-end delay. As can be seen in figure 3, the LSP from FDA to FA1 will pass through FA3, which is also a LER and can play the same role as FDA or old FA. So FA2 can not only receive packets from FDA or FA1, but also receive packets from FA3.

As can be seen in figure 3, the key problem is to decide the LER, FDA or FA1 or FA3, which the LSP passes through. The paper believes that; when MN is not receiving packets, MN should send Registration Message to FA, then FA send the message to FDA. At the same time, the previous established LSP should be released.

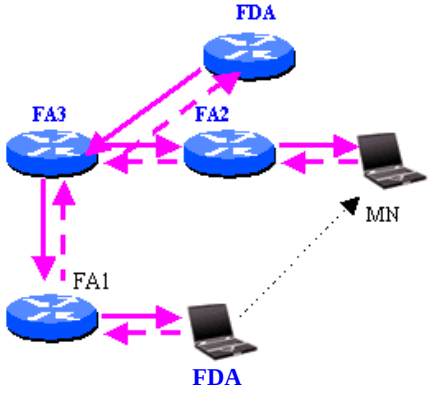


Fig. 3. The architecture of DTAHMM

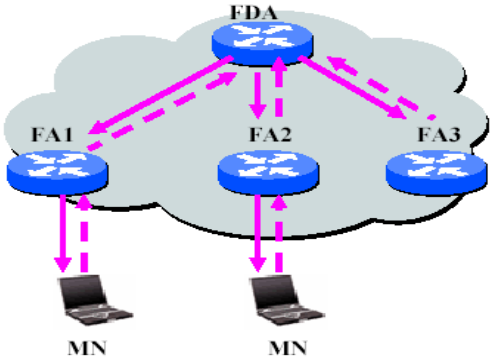


Fig. 4. H-MPLS architecture

However, when MN receives packets and handoff happens. As shown in figure 3,when MN moves from FA1 to FA2, the procedure can be described in the following.

(1) Before MN detaches its current link with FA2, the MN will receive some Layer 2 message, which provides the information of the new prefix of subnet. The MN may use stateless address auto-configuration to form a new CoA.

(2) After MN forms its new CoA, it will send a PRE_REG (pre-registration), which includes IP address of FA2, MN's new CoA, and MN's old CoA to its FA1.

(3) Being receiving PRE_REG, FA2 multicasts an LSP_TUNNEL_REQ (LSP tunnel request) to the other LERs in the same MPLS domain. These LERs is limited to the LER on the LSP used by MN. So all the other LERs send an LSP_TUNNEL_REQ to FA2.

(4) Having received LSP_TUNNEL_REQ, FA2 adds an entry for the MN that binds the MN's old CoA with new CoA, and sends back LSP_TUNNEL_REP (LSP tunnel reply) to the sender, such as FA3, whose message is firstly received by FA2. And FA2 will leave the later LSP_TUNNEL_REQ messages. When such sender receives LSP_TUNNEL_REP, a bi-direction LSP tunnel between FA2 and such sender has been established. At this time, such sender is a temporary agent that plays the role of mobile access point.

(5) MN sends a binding update to such temporary agent. When this temporary receives the BU from MN, it will send a BUACK to MN.

(6) FDA will use the newly established LSP to send packets to FA3, then to FA2.

(7) FA2 will send an LSP_TUNNEL_REL (LSP tunnel release) message to FA3, then FA3 send such message to FDA.

(8) Being receiving LSP_TUNNEL_REL, FDA sends a LSP_REL_REP (LSP release reply) to the FA3 and then sends it to FA2.

5. PERFORMANCE ANALYSIS

In this section, we will discusses the performance of such scheme. Let's analyze the end-to-end delay. In order to analyze the delay, the paper expresses a sign in the following.

T_{AtoB} : The processing and propagation delay of a packet to be transferred through a LSP that is from A to B. For example, $T_{FA1toFA2}$ denotes the processing and propagation delay of a packet to be transferred from FA1 to FA2 in figure 3.

In figure 3, it is easy to understand $T_{FDAtoFA2} = T_{FDAtoFA1} + T_{FA1toFA2}$. As $T_{FDAtoFA1} = T_{FA1toFDA}$. According AHMMS, we can evaluation the end-to-end delay of AHMMS as following.

$$\begin{aligned} &Delay\_of\_AHMMS \\ &= \min\{T_{FDAtoFA1} + T_{FA1toFA2}, T_{FDAtoFA2}\} + T_{CNtoFDA} \\ &= \min\{T_{FDAtoFA3} + T_{FA3toFA1} + T_{FA1toFA2}, T_{FDAtoFA2}\} + T_{CNtoFDA} \end{aligned} \quad (1)$$

On the other hand, when we come to DTAHMM, we can calculate the delay of it as the following.

$$\begin{aligned} &Delay\_of\_DTAHMM \\ &= \min\{T_{FDAtoFA1} + T_{FA1toFA2}, T_{FDAtoFA2}, T_{FDAtoFA3} + T_{FA3toFA2}\} + T_{CNtoFDA} \\ &= \min\{T_{FDAtoFA3} + T_{FA3toFA1} + T_{FA1toFA2}, T_{FDAtoFA2}, T_{FDAtoFA3} + T_{FA3toFA2}\} + T_{CNtoFDA} \end{aligned} \quad (2)$$

Let's compare (1) and (2). Obviously, $Delay\_of\_DTAHMM \le Delay\_of\_AHMMS$.

Table 1. Each values of TAtoB which is shown in Fig. 2.

T_{AtoB}	$T_{CNtoFDA}$	$T_{FDAtoFA3}$	$T_{FA1toFA3}$	$T_{FA2toFA3}$	$T_{FA1toFA2}$	$T_{FDAtoFA2}$
Value(ms)	800	160	240	80	100	520

According to (1):

$Delay\_of\_AHMMS = 1300ms$,

According to (2):

$Delay\_of\_DTAHMM = 1040ms$

From the result we also can conclude $Delay\_of\_DTAHMM \le Delay\_of\_AHMMS$.

The handoff delay is the period that MN switches from old FA to new FA. In DTAHMM, the temporary agent is the agent that new FA first receives LSP_TUNNEL_REQ message, so the handoff delay must shorter than AHMMS.

6. CONCLUSIONS

In this paper, according to the Reformative Hierarchical MPLS (RH-MPLS) and the Adaptive Hierarchical Mobile MPLS Scheme (AHMMS), we propose a Dynamic Temporary Agent In Hierarchical Mobile MPLS (DTAHMM). In AHMMS the key problem is the selection of LER in a H-MPLS domain. In fact, the LSP in a domain usually includes at least two LER, which LER is the most suitable is necessary to be discussed. So the paper proposes a DTAHMM. The performance analysis shows that DTAHMM is somehow better than AHMMS.

7. REFERENCES

[1] D. Johnson, "Mobility support in IPv6", draft-ietf-mobileip-ipv6-19.txt, October 2002.

[2] E. Rosen, "Multiprotocol label switching architecture", RFC3031, January 2001.

[3] C. Perkins, "IP Mobility Support", RFC2002, October 1996

[4] R.Zhong, Chen-Khong Tham, et al, "Integration of Mobile IP and MPLS", Internet Draft, July 2000

[5] H. Soliman, "Hierarchical Mobile IPv6 mobility management", draftietf-mobileip-hmipv6-07.txt, October 2002.

[6] T.Yang, D.Makrakis, "Hierarchical Mobile MPLS: supporting delay sensitive applications over wireless internet", International Conference on Info-Tech & Info-Net (ICII 2001), Beijing, China, October 2001.

A Novel Cross-layer Quality-of-service Model for Mobile Ad hoc Networks*

Leichun Wang, Shihong Chen, Kun Xiao, Ruimin Hu
National Engineering Research Center of Multimedia Software, Wuhan University
Wuhan 430072, Hubei, China
Email:wlc2345702@163.com

ABSTRACT

The divided-layer protocol architecture for Mobile Ad hoc Networks (simply MANETs) can only provide partial information exchange for different layers in the protocol stack. This leads to great difficulties in QoS guarantee of multimedia communication in MANETs. To improve QoS of multimedia information transmission in MANETs, this paper proposes A Cross-layer QoS Model for MANETs, CQMM. In CQMM, a core component was added, Network Status Repository (NSR), which was the center of information exchange and share among different protocol layers in the stack. At the same time, CQMM carried out all kinds of unified QoS controls. It is advantageous that CQMM avoids redundancy functions among the different protocol layers in the stack and performs effective QoS controls and overall improvements on the network performances.

Keywords: Cross-layer; QoS Model; Mobile Ad hoc Networks (MANETs); Network Status Repository (NSR); QoS Controls.

1. INTRODUCTION

With the rapid development of multimedia technologies and the great increase of the bandwidth for personal communication, Voice and Video services begin to be deployed in MANETs. Different from static networks and Internet, multimedia communications in MANETs such as Voice and Video services require strict QoS guarantee, especially the delay guarantee. In addition, communication among different users can be integrated services with different QoS requirements. These lead to great challenges in QoS guarantee of multimedia communication in MANETs. There are two main reasons in these: 1) MANETs runs in a typical wireless environment with time-varying and unreliable physical link, broadcast channel, and dynamic and limited bandwidth and so forth. Therefore, it can only provide limited capability for differentiated services with strict QoS requirements [1]. 2) It is difficult that traditional flow project and access control mechanism are implemented because of mobility, multiple hops and self-organization of MANETs.

At present, most researches on QoS based on traditional divided-layer protocol architecture for MANETs focus on MAC protocol supporting QoS [2], QoS routing protocol [3] and adaptive application layer protocol with QoS support [4], and so on. It is avoidless that there will be some redundancies on functions among the different protocol layers in the stack. This will increase the complexity of QoS implementation and cause some difficulties in overall improvement on the network performances. Therefore, it is not suitable for MANETs with low processing ability.

In recent years, the cross-layer design based on the partial protocol layers in MANETs was put forward. [1] proposed the mechanism with QoS guarantee for heterogeneous flow by information exchange between the network layer and the MAC layer. [5, 6, 7, 8] did some researches on implementing video communication with QoS guarantee by exchange and cooperation of information among a few layers in MANETs. These can improve QoS in MANETs' communication to some extent. However, MANETs is much more complex than wired system and static network, and improvements on QoS guarantee depend on full cooperation among all layers in the protocol stack. Therefore, it is difficult for the design to provide efficient QoS guarantee for communication and overall improvements on the network performances in MANETs.

To make good use of limited resources and optimize overall performances in MANETs, this paper proposes a novel cross-layer QoS model, CQMM, where different layers can exchange information fully and unified QoS managements and controls can be performed.

The rest of the paper is organized as follows. CQMM is described in section 2 in detail. In section 3, we analyze CQMM by the comparison with DQMM. The section 4 concludes the paper.

2. A CROSS-LAYER QOS MODEL FOR MANETS —CQMM

2.1 Architecture of CQMM

In MANETs, present researches on QoS are mostly based on traditional divided-layer protocol architecture, where signals and algorithms supporting QoS are designed and implemented in different layers respectively, such as MAC protocol supporting QoS in data link layer [9], routing protocol with QoS support in network layer [10,11], and so forth. It can be summarized as A Divided-layer QoS Model for MANETs, DQMM (see Fig. 1.).

In DQMM, different layers in the protocol stack are designed and work independently; there are only static interfaces between different layers that are neighboring in logic; and each protocol layer has some QoS controls such as error control in logic link layer, congestion control in network, etc. On the one hand, DQMM can simplify the design of MANETs greatly and gain the protocols with high reliability and extensibility. On the other one, DQMM also has some shortcomings: 1) due to the independent design among the different protocol layers, there are some redundancy functions among the different protocol layers in the stack. 2) it is difficult that information is exchanged among different layers that are not neighboring in logic, which leads to some problems in unified managements, QoS controls and overall improvements on the network

* National Natural Science Foundation of China supported the work under grant No. 60472047.

performances.

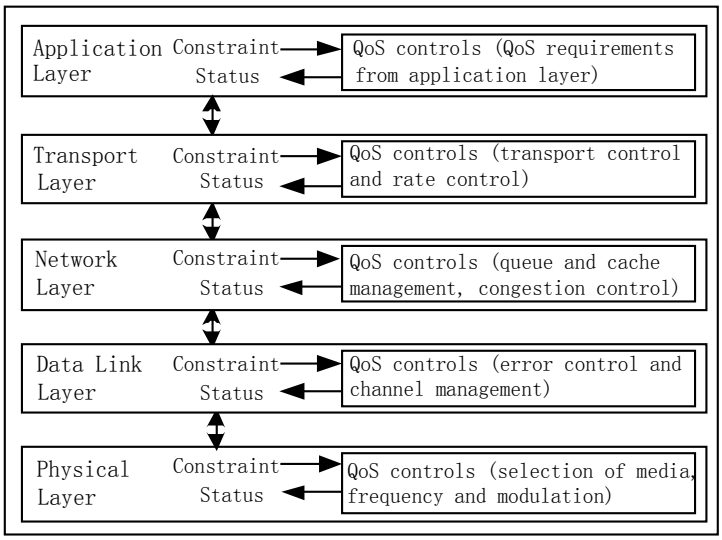


Fig. 1. Architecture of a divided-layer QoS model for MANETs, DQMM

Therefore, it is necessary that more attention are focused on the cooperation among physical layer, data link layer, network layer and higher layers when attempting to optimize performances of each layer in MANETs. For this reason, we combine parameters dispersed in different layers and design a novel cross-layer QoS model, CQMM, to improve the QoS guarantee and the overall network performances. The architecture of CQMM is provided in Fig. 2..

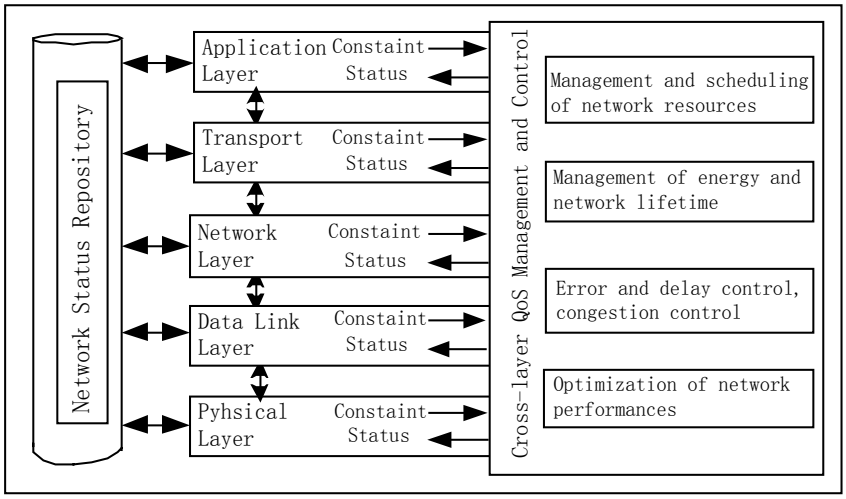


Fig. 2. Architecture of a cross-layer QoS model for MANETs, CQMM

From Fig.2, it can be seen that CQMM keeps the core functions and relative independence of each protocol layer in the stack and allows direct information exchange between two neighboring layers in logics to maintain advantages of the modular architecture. On the basic of these, a core component is added in CQMM, Network Status Repository (simply NSR). NSR is the center, by which different layers can exchange and share information fully. On the one hand, each protocol layer can read the status information of other protocol layers from NSR to determine its functions and implementation mechanisms. On the other one, each protocol layer can write its status information to NSR that can be provided with other layers in the protocol stack. In CQMM, the protocol layers that are neighboring in logics can exchange information directly or indirectly by NSR, and the protocol layers that are not neighboring in logics can exchange information using cross-layer ways via NSR. Therefore, information exchange is flexible in CQMM.

All kinds of QoS controls in CQMM such as management and scheduling of network resources, network lifetime, error control, and congestion control and performance optimization and so on, are not carried out independently. On the contrary, CQMM is in charge of the unified management and all QoS controls by the cooperation among different protocol layers in the stack. Each QoS control in MANETs is related to all layers in the protocol stack, and also constrained by all layers in the stack. The results of all QoS operations and managements are fed back to the different layers and written back to NSR, which will become the parameters of all kinds of QoS controls in MANETs.

2.2 Protocol Design in CQMM

In CQMM, the protocol designs aims at the full and free information exchange and cooperation among different protocol layers to avoid possible redundancy functions when

maintaining the relative independence among different layers and the advantages of the modular architecture.

Physical Layer: Physical layer is responsible for modulation, transmission and receiving of data, and also the key to the size, the cost and the energy consumption of each node in MANETs. In CQMM, the design of physical layer is to choose the transmission media, the frequency range and the modulation algorithm with the low cost, power and complexity, big channel capability and so on, according to the cost of implementation, energy constraint, and capability and QoS requirements from high layers.

Data link layer: The layer is low layer in the protocol stack and can be divided into two sub-layers: logic link sub-layer and MAC sub-layer. Compared with high layers, data link layer can sense network status in MANETs earlier such as the change of channel quality, the network congestion and so on. Therefore, on the one hand data link layer can perform the basic QoS controls such as error control and management of communication channel. On the other one, the layer can be combined with high layers to establish, choose and maintain the routing faster, prevent the congestion of the network earlier, and choose appropriate transport mechanisms and control strategies for transport layer.

Network Layer: The design and implementation of network layer protocol in CQMM is to establish, choose and maintain appropriate routings by taking into consideration the power, the cache, the reliability of each node in a routing, QoS requirements of services from high layer such as the bandwidth and the delay, and implementation strategies of low layers such as the mechanism of error control in logic link sub-layer and the way of the channel management in MAC sub-layer.

Transport Layer: In CQMM, the protocol design of transport layer needs to be aware of both functions and implementation mechanism of lower layers such as the way of error control in data link layer, the means to establish, choose and maintain routing in the network layer, and QoS requirements from the application layer, to determine corresponding transmission strategies. In addition, the transport layer also needs to analyze all kinds of events from low layers such as the interrupt and change of the routing and the network congestion, and then respond properly to avoid useless sending data.

Application Layer: There are two different strategies in the design of the application layer: 1) differentiated services. According to the functions provided by the low layers, applications are classed as the different ones with different priority levels. 2) Application-aware design. Analyze specific requirements of different applications such as the bandwidth, the delay and the delay twitter and so on, and then assign and implement the functions for each layer in the protocol stack according to the requirements.

2.3 QoS Cooperation and Management in CQMM

In CQMM, the core of QoS cooperation and management is that NSR acts as the exchange and share center of status information in protocol stack, and by the full exchange and share of network status among different protocol layers the management and scheduling of the network resources and the overall optimization of the network performances can be implemented effectively. They include the management and scheduling of the network resources, the cross-layer QoS cooperation and control, and the overall optimization of the network performances.

Management and Scheduling of Network Resources: Network resources include all kinds of resources such as the cache, the energy and the queue in each node, and the communication channel among nodes and so forth. In CQMM, the management and scheduling of the network resources are not to meet the requirements of the different layer respectively, but to attain the unified management and scheduling of the network resources and full utilization of limited resources in order to increase the QoS of all kinds of communication.

QoS Cooperation and Control: In CQMM, all kinds of QoS controls and cooperation such as the rate adaptation, the delay guarantee and the congestion control and so on, are not implemented by each layer alone, but completed through the operation of all layers in the protocol stack. For example, the congestion in MANETs can be earlier prevented and controlled by the cooperation among different layers such as ACK from MAC sub-layer, the routing information and the loss rate and delay of package from network layer, and the information of rate adaptation in transport layer and so on.

Performances Optimization: In CQMM, the optimization of the network performances aims to establish a network optimization model constrained by all layers in the protocol architecture and finds the “best” ways according to the model in order to improve the overall performances in MANETs.

3. ANALYSIS OF CQMM

Present QoS models for MANETs can mainly be classed as a QoS model based on traditional divided-layer architecture DQMM and a cross-layer QoS model proposed by this paper CQMM. QoS model used by [1,5-8] is to some extent extended on the basis of DQMM in nature. Here, we only compare CQMM with DQMM.

3.1 Information Exchange

Different protocol architecture and principle between CQMM and DQMM lead to great differences in the means, the frequency, the time and the requirement of the information exchange. (see Table 1.)

From Table 1., it can be seen that compared with DQMM CQMM has some advantages: 1) more flexible information exchange. Neighboring layers can exchange information by the interfaces between layers or NSR, and crossing layers may exchange information through NSR; 2) simpler transform in information format. Different layers can exchange information by NSR, so these layers only need to deal with the format transform between the layers and NSR; 3) lower requirements. The protocol layers can read them in proper time Information from different protocol layers temporarily stored in NSR, so the layers exchanging information are not required to be synchronous in time; 4) more accurate control. NSR in CQMM can store information of some time from the different layers, which is advantageous to master the network status and manage the network more accurately. However, these require higher information exchange frequencies among the different layers, more processing time of each node, and more communication among them.

Table 1. Comparisons of information exchange between CQMM and DQMM

Comparison of information exchange	DQMM	CQMM
Means	Neighborhood layers:static interfaces between layers Cross-layer: middle layers	Neighborhood layers:static interfaces or NSR Cross-layer: NSR
Frequencies	neighborhood layers: high crossing layers:low	Neighborhood layers:high Cross-layer: high
Format transform	neighborhood layers:once Cross-layer: many times	Format transform between each layer and NSR
Participant layers	All Participant layers and middle layers	NSR and Participant layers
Time requirement	Specific time	“Some time”
Synchronization reqiurement	Severe synchronization	Asynchronism
Exchange time	Little	Much
Information size	Neighborhood layers:big Cross-layer: small	Neighborhood layers:big Cross-layer: big

3.2 Protocol Design

In DQMM, it is inevitable that there are some redundancy functions among different protocol layers for implementing reliable information transmission because of the independence in function and protocol design. However, CQMM can perform unified function assignation among different layers and implement communication with QoS support by the cooperation among the different layers.

3.3 Management and Scheduling of Network Resources and Performances Optimization

The lack in full information exchange among different protocol layers in the stack limits the overall improvements on the network performances in DQMM. In CQMM, different layers in the protocol stack can exchange information freely and fully through NSR, which benefits unified planning and scheduling of the network resources, establishing performance optimization model based on the whole network, and performing the overall improvements on the network performances.

3.4 Cost and Complexity of Implementing Protocols

Due to the independence of each protocol layer in the stack, there are less interfaces and information exchange among different layers, so DQMM needs lower cost and complexity. In CQMM, free and full information exchange among different protocol layers of the stack requires: add NSR in each node to store network status and provide replacing strategies to update status information in time; increase interfaces where each protocol layer and NSR exchange information; establish more complex mathematics model and control mechanism to optimize network performances. These lead to the increase of the implementation’s cost and the complexity in CQMM.

Compared with DQMM, CQMM can provide free and full information exchange for different layers in the protocol stack, eliminate redundancy functions among the different layers, better realize the management and scheduling of the network resources, and optimize overall performances in MANETs. These advantages are at the cost of higher cost and complexity.

4. CONCLUSIONS AND FUTURE WORK

This paper provides a new cross-layer QoS model for MANETs, CQMM. Compared with DQMM, CQMM has the following advantages: 1) different layers in the protocol stack can exchange information freely and fully through NSR; 2) CQMM can eliminate redundancy functions among the different layers in the protocol stack, and better implement unified management and scheduling of the network resources, and the overall improvements on the network performances. On the basis of CQMM framework, future work will focus on the design of the cross-layer protocols, the cross-layer control and cooperation of QoS, and the overall improvements of the performance of MANETs using cross-layer methods.

5. REFERENCES

[1] L.Wei, X.Chen et al, “Courtesy piggybacking: supporting differentiated services in multihop mobile ad hoc networks”, Mobile Computing, IEEE Transactions, 2004, 3(4): 380-393.

[2] G.Ahn, A.T. Campbell, et al, “Supporting Service Assu- rances in Multihop Wireless Local Area Networks”, IEEE Global Telecommunications Conference 2003, San Francisco, CA, USA, Dec. 2003.

[3] G.Mohsen, Wireless Communications Systems and Networks, Kluwer Academic Publishers, 2004.

[4] G.Andrea, Wireless Communications, Cambridge University Press, 2004.

[5] W.Kumwilaisak, Y.T.Hou, et al, “A Cross-layer Quality-of-service Mapping Architecture for video Delivery in Wireless Networks”, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, 2003, 21(10).

[6] Eric Setton, T. Yoo, et al, “Cross-layer design of ad hoc networks for real-time video streaming”, Wireless Communications, IEEE, 2005,12(4): 59-65.

[7] Q. Qu, Y. Pei et al, “Cross-Layer Design for Delay-Constrained Error-Resilient Video Communications Over Wireless Ad-Hoc Networks”, Image Processing, ICIP 2005, 2005, Vol.3: 720-723.

[8] Taesang Yoo, Eric Setton et al, “Cross-layer design for

video streaming over wireless ad hoc networks", Multimedia Signal Processing, 2004 IEEE 6th Workshop, 2004.

[9] Adopted by the ISO/IEC and redesignated as ISO/IEC 8802-11:1999(E).

[10] Raghupathy Sivakumar, Prasun Sinha, et al, "CEDAR: A Core-Extraction Distributed Ad hoc Routing Algorithm", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, 1997, 17(8): 1454-1465.

[11] S.G.Chen, Klara Nahrstedt, "Distributed Quality-of-Service Routing in Ad hoc Networks", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, 1997, 17(8): 1488-1505.

Peer-to-Peer Approaches to Query Processing in Internet-scale Networks

Weining Qian
Department of Computer Science and Engineering
Fudan University, Shanghai 200433, China
Email: wnqian@fudan.edu.cn

And

Aoying Zhou
Department of Computer Science and Engineering
Fudan University, Shanghai 200433, China
Email: ayzhou@fudan.edu.cn

ABSTRACT

Query processing in Internet-scale networks poses challenges such as fully decentralized processing, instability of nodes and network, and heterogeneity of nodes. Though peer-to-peer (P2P) technologies have been widely adopted in Internet-scale applications, they are far from meeting the requirements for complex query processing in efficiency.

In this paper, the challenges for designing a query processing engine over P2P network are discussed first. Then, some general considerations about the P2P-based query processing system and the underlying ideas are presented. Based on these considerations, two approaches, say, pull-based and push-based query processing, are presented. Meanwhile, two prototyping systems, P-Terse and Sonnet, are introduced as efforts for implementing P2P query processing system with the two approaches, respectively. The former is a peer-to-peer based text retrieval and search system, and the latter is a subscription overlay network. This paper is a brief summary of the research and development work, which have been done in our research group.

Keywords: Peer-to-Peer computing, query processing, information retrieval

1. Introduction

Peer-to-Peer (P2P) systems have emerged in Internet-based applications, such as file sharing and instant message passing. Nowadays P2P-based systems have been becoming the hot topics in both academia and industrial communities. Existing efforts on research and development of P2P-based technologies include the design and analysis of overlay network structures (and corresponding routing algorithms) and the implementation of advanced functionalities such as distributed storage, retrieval, and computation, over P2P networks.

Query processing is a natural extension to match- based search provided by existing file-sharing applications. It enables searching for the information in a finer granularity compared to the search for files based on file names. It is a critical technology for both information retrieval and database management applications.

To the best of our knowledge, it could be claimed that general-purpose P2P technologies are not sufficient for providing query processing in a large-scale distributed network as Internet. There are several challenges to implement query-processing functionalities in a large-scale network:

Heterogeneity of nodes: Nodes are equipped with different hardware and network connections. This should be considered by any P2P systems. The heterogeneity of nodes poses additional challenges to query processing. Since the data stored on each node may have different schema, the method for searching or retrieving data should not be based on schema. Furthermore, no any assumption about the storage media and/or access methods of data could be made a priori. In other words, the interface available on each node to be used by the query processing is simpler than those in centralized or conventional distributed databases.

Expressive power of queries: Query processing is more complex than keyword-match-based file name search. The data to be queried is provided in a certain schema. The queries should be sufficiently powerful for retrieving the data under that schema. There are three main challenges related to the expressive power of queries. Firstly, there are several widely used data types. The valid operators are different for different data types. For example, range query is only available for numeric data types but not for categorical data type. However, most general-purpose P2P-based technologies only consider categorical data, which strictly limit the application of them in complex query processing. Secondly, schema information itself is the data to be queried. In centralized or conventional distributed databases, schema information is maintained in system catalog and is available to the query processing modules. It is known that P2P systems are fully decentralized. Therefore, the maintenance of schema information should be distributed to the network, and access to this information should be efficient. Last but not the least, different applications have different data model. Information retrieval, for example, takes the vector-space model, which is essentially a multi-dimensional data model. Meanwhile, instant messaging applications usually use XML as data model. The question is whether we should use different overlay networks for each different data model or use only one overlay network to support all of them, or some any other approach in between.

Efficiency and scalability: The main concern of traditional query processing is on the performance when handling a large volume of data. Large-scale network poses additional challenge on efficiency. The system is required to keep efficient when the number of nodes increases. This means, firstly, few data transmitting over the network should be conducted for evaluating one query, so that low-latency response times are achieved. Secondly, the overhead for maintaining the P2P system should be small enough so that it would not affect the efficiency of query processing. The overhead consists of two parts, one

is for the maintenance of the network, and the other is for keeping the data available and consistent. Furthermore, the workload should be balanced, that is to say, the load assigned on each node should not exceed its capacity.

Approximate query processing: P2P networks usually are large-scale networks which comprise thousands or even more nodes. To search the whole network to address a user-given query is time consuming, and so it is intolerable from the point of view of users or systems, even it is not impossible. Therefore, in most cases, a user may only get a partial result for such a specific query. Thus, it is necessary to develop some intelligent search and ranking techniques for query processing in P2P systems, which is rarely considered in centralized or conventional distributed databases.

Facing the challenge mentioned above, two approaches for query processing, say, pull-based approach and push-based approach, are proposed in this paper to address the important issues we might encounter in developing query-processing engine over large-scale networks. Moreover, two prototyping systems are deigned and implemented to verifying the underlying ideas of these two approaches. Some technical detail of the two systems are described and discussed in this paper.

The remained parts of this paper are organized as follows. In Section 2, some general considerations for designing a query processing system over a large-scale network are discussed. The pull-based and push-based approaches and the prototype systems and corresponding prototyping systems are presented in Section 3 and 4, respectively. Finally, Section 5 is for concluding remarks and discussion about further research work.

2. General Considerations on P2P Query Processing

We present our basic considerations for designing a P2P-based query processing system in this section. The intuition behind the considerations and the analysis of the advantages and disadvantages of these considerations are discussed in details.

Multi-tier architecture

P2P-based systems are essentially network-based application systems. A multi-tier architecture is a natural choice for such kind of systems. Currently, there are some different P2P network structures, which are proposed for achieving different features of the P2P systems. For example, structured P2P network is invented for achieving fast routing, while unstructured P2P network could be employed to organize nodes based on similarity of information over peers. Neither of these two P2P network structures can meet the requirements from complex applications such as query processing systems. Therefore, a possible way is to design a delicate hybrid structure to combine the respective advantages of the structured and unstructured P2P networks elegantly.

The multi-tier architecture is analog to the structure of traditional query processors. The overlay network serves as the underlying storage and data access method. The higher tiers are the query operators, query optimizer, and query interface. On the other hand, it is shown that there exists a major difference between this multi-tier architecture and the conventional multi-tier query processors, that is, not only the overlay network is spanning over multiple nodes, the query operators and query optimizer should also be distributed. Thus, P2P technologies should be used in all these tiers.

Another advantage of this architecture is that the interfaces can be defined clearly. Furthermore, it is flexible for further extension.

Overlay network as a service

It is suggested in this paper that only one overlay network is used for as many applications as possible, because the maintenance of overlay networks will result in additional overhead on storage, computation, and network bandwidth. Since most overlay networks provide key-value lookup function, it is not necessary to introduce another overlay network for a new application.

Overlay network can be treated as a service that provides key-based access to data. Several systems take this approach [1, 2, 3]. However, being different from centralized access methods such as B-tree index, with this approach system may need to access remote data in storage or retrieval sometimes, and it is quite expensive when compared with local data access. Therefore, it should not be overused. The method of mapping between data and nodes should be deliberately designed. Meanwhile, additional optimization methods should be introduced to further reduce the response time and network bandwidth consumption.

Another important issue is to determine the functions that should be provided by the overlay network. We should keep the interface of overlay network powerful enough for query processing, while not pollute it by pushing too many functions beneath the interface. Therefore, the structure of the overlay network should be carefully designed and the functionalities of it should be deliberately chosen.

In-network processing

One of the advantages of P2P systems is that physically distributed peers may collaborate with each other to perform a complex task. It is especially helpful for complex query processing, since the data are distributed and each node usually has spare computation power that could be used by the query processor. Therefore, the so-called in-network processing is recommended here, which means each node may evaluate some operators over local data, and then the results derived from each individual local data could be aggregated to generate the final answer. So, such a process consists of two phases, remote processing phase and result integration phase.

It is important to determine an elegant interface for the remote processing. On the one hand, the remote nodes should not be overloaded. On the other hand, the results from remote nodes should be easy to be merged. Furthermore, it is ideal if data is processed while it is transmitted. By doing so, the result merging could be also distributed over remote nodes.

Taking all the above into account, a conceptual architecture for P2P-based query processor as shown in Fig. 1. is proposed in this paper. The communication management acts as a storage and access method service at the bottom of the system, while the local storage acts as a similar role. The overlay network is equipped with a data and query-mapping module, which translates the database operators into network primitives. The mapping module separates the overlay network and the local query processor. Different mapping schemes correspond to different kinds of query processing tasks. They can share

the same overlay network structure. The rest parts of the architecture are just the same as a conventional query processor. To support in-network processing, the data received by the communication management module is returned to the local processing module, and then it is sent back to the overlay network, so that each node in the system can participate in the processing.

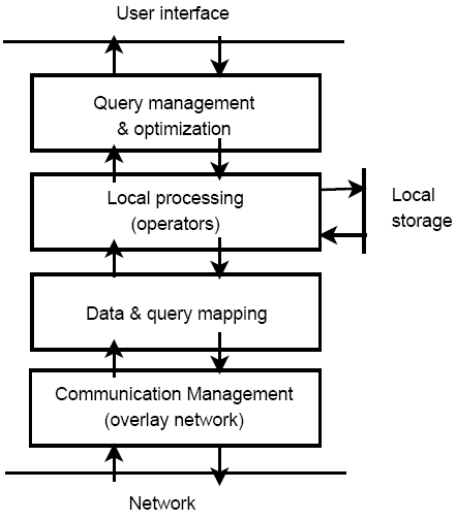


Fig. 1. A conceptual architecture of a P2P query processor

3. Pull-based Query Processing

If a system retrieves and returns the result for a query when receiving a query request, such a system is called a *pull-based* query processing system. A system is called a *push-based* system if it outputs query results continuously to the user after a query is registered. The pull-based P2P query processing is first introduced in this section.

In a P2P system, the pull-based query processor has to traverse the network to visit all the nodes with answers to user's query. The results of the remote nodes are transmitted in the network while being merged to form the final answer to the query. Thus, a typical query processing session is as follows:

1. The querying node determines the initial list of nodes to be searched;
2. The query is forwarded to the nodes in the list;
3. Each node which received the query evaluate the query based on its local database;
4. The node also generate a new list of nodes to be searched further, if the list is not empty, go to step 2;
5. The node getting the local result may send the result back to the querying node, or wait for results from other nodes, integrate them with local result, and send it to another node that can integrate the result further. This process stops when the result arrives at the querying node.

Step 1 to 4 is in to the query-forwarding phase, while the step 5 is the result-collecting phase.

In such a session, the search method, i.e. the method for determination of the node list, is especially important for performance. The method should find the most informative nodes without flooding the query to all nodes. Basically, there are two approaches to search in P2P systems, *state-aware* methods and *state-free* methods. The state-aware methods maintain the index of the data in the system. Thus, after the index entries related to the query are accessed, a node has collected sufficient information to search the network efficiently without much unnecessary overhead. However, this method is quite expensive in terms of index maintenance. The key-based Distributed Hash Table (DHT) based methods fall into this category. On the contrary, in state-free methods, there is no node for storing global information. Each node determines the node list just based on its limited local information. Sometimes the local information is even missing. For example, naïve yet widely adopted methods like breadth-first search do not require the node to provide any information except for a list of neighboring nodes. Other methods falling into this category include BestPeer [4] and SHINOV [5]. BestPeer employs a self-configurable network maintenance mechanism, while SHINOV a heuristic-based searching mechanism which takes the small-world phenomenon into consideration [4,5].

A complete query processing process may involve more than one query processing session. For example, in PeerDB, a BestPeer-based structured query processing engine, the first session is dedicated to the retrieval of the schema information, while the second session is for the retrieval of results [6]. Similarly, in our P-Terse prototype system, a P2P-based text retrieval system, the first session is for the retrieval of keyword index entries [7]. The second session is for the retrieval of top-*k* document list, while the third session is for the retrieval of the documents based on user's feedback on the list. Since the characteristics of query and result are different, different technologies may be used in sessions of one query processing process.

P-Terse: Peer-to-peer based TExt Retrieval and SEarch system

P-Terse is a prototyping system developed in our research group as a testbed for various P2P technologies [7]. It is a pull-based text search and retrieval system. Simulation experiments over thousands of nodes and real test over tens of nodes are conducted to verify the efficiency and effectiveness of the proposed methods and architecture. A brief description of the system is presented here to illustrate the design principles introduced above. More details about the system can be found in some previously published papers and technical reports such as [5, 7, 8, 9, 10, 11].

The architecture of the node is shown in Fig. 2.. It is a standard multi-tier structure. Note that there are two tiers devoted for searching. One is the overlay communication module, which is for node identifier based search. The other one is the search module that is for the content-based searching. The overlay network is a state-aware tier, which keeps the network connected. It also provides the basic support for index-based searching.

Our initial implementation of the search module takes a state-free approach [7]. A heuristic search method, called SHINOV [5], is deigned and employed in the system. Each node keeps the nodes with similar content as its neighborhood. To evaluate a keyword query, it is only forwarded to a small portion of neighboring nodes if the query is not similar to the node's local content. When the query arrives at a node whose content is similar to the query, it is broadcast to all neighboring nodes. Intuitively, this search schema would throw the query into the cluster of nodes containing much content related to the query.

Clusters of nodes are generated by self-configurable network, and the nodes with similar content are put in the same cluster. Therefore, little network bandwidth is wasted to forward queries to nodes having no result. Both the node clustering and the querying process are distributed. Furthermore, each node does not need to store any information except the neighborhood list.

SHINOV works well in our simulation experiments and real-life testing with tens of nodes where the resulted documents do not to be sorted in the list, and the accuracy is measured by a Boolean function, say, related-to-the-query function. To support more complex text retrieval models such as vector space model, a method for indexing the keywords called SIPPER [10] is studied as well. It takes a state-aware approach. The statistic information of the keywords is maintained using DHT. For query processing, the statistic information is collected. Then the nodes are ranked based on the similarity between their content and the query. Informative nodes are ranked higher than other nodes. Only the top ten nodes in the ranked list are visited for retrieving the final results. However, empirical results show that this method can achieve quite high precision and recall. This greatly saves the network bandwidth consumption, while keeping the search accuracy being high. With this method, a more accurate document ranking could be achieved [8]. Though the optimization method for workload balancing is employed [11], the system sometimes still suffers the expensive cost for index maintenance. Thus, more effective state-aware method for content-based search needed to be explored.

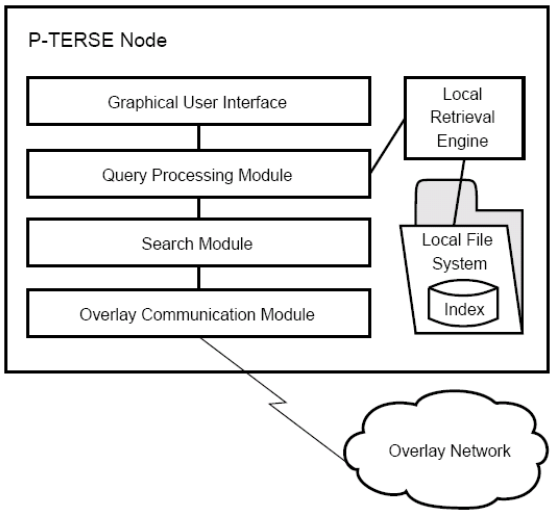


Fig. 2. The architecture of a P-Terse node.

The query processing module in P-Terse is essentially a schedule and session management tier. It translates user's queries to the calls to the search functions provided by the search and overlay tier. Furthermore, it manages the multiple sessions and stores the intermediate results. The local retrieval engine is implemented based on an open source retrieval engine, Lucene. It monitors and indexes documents stored in local file system that are shared by the user. We are working on more technologies that can be implemented in the system, such as user control, document versioning, and access control.

Other efforts on P2P pull-based query processing

There are quite a lot of P2P pull-based query processing projects, such as PeerDB [6], PIER [1], PlanetP [12], and PSearch [13], to name a few.

PeerDB [6] is actually a query-processing engine based on BestPeer [4], an unstructured P2P platform. PeerDB uses a two-phase (session) process for query processing. In the first session, an information retrieval process is conducted for retrieving the schema information, while in the second session, actual data are queried. Furthermore, it uses self-configuration network to optimize the search.

PIER [1] is a database query processing system based on structured overlay network. The overlay network is used as a basic indexing scheme. Query operator implementation and optimization techniques are studied.

Different from PeerDB or PIER, PlanetP [12] and PSearch [13] are designed for information retrieval tasks. The former is based on unstructured P2P network, and the later is based on structured one. Efficient search algorithms, and keyword indexing schemes are studied in depth.

4. Push-based Query Processing

Push-based query processing is essentially processing for continuous query. The query is relatively static, while the data is dynamic. Therefore, the system should not only evaluate the query, but also *manage* the queries. Meanwhile, the scheme for evaluate queries should be modified. The data (result) should actively seek for the queries it can answer over the network. Thus, a common push-based query processing session is as follows.

1. The querying node registers its query in the system;
2. When new data is generated (or data is updated), the node that owns the data searches over the network for related queries;
3. The data is sent to the nodes with the related queries;
4. Step 2 and 3 iterates until a previously defined stop condition is reached, or there is not nodes containing the related query anymore.

Here, Step 2 is actually a pull-based query processing session. This step should be implemented rather efficiently, since it is invoked frequently in a push-based query processing. Therefore, it is suggested to embed this function into the overlay network to taking full utilization of the fast routing mechanism provided. Another important issue is about the management of queries. Because we do not want any node lost potential results of its queries, an index-based method could be employed. Thus, push-based query processor favors state-aware methods. Overlay networks play a very important role in such systems.

The third step is also interesting in push-based query processing. Though sending the result to each querying node is simple to implement, the node that holds the result may suffer the overload problem when so many nodes query the holding data. Therefore, this result forwarding process should be distributed over the network. Sometimes, the result of one query is the partial result of another query. Then, the data can be processed while being transmitted in the network, so that network bandwidth can be saved, while the computation power of intermediate nodes can be fully utilized. Since this in-network processing scheme relies on the relationships of the queries, it is suggested to use state-aware method to maintain the relationships of the nodes (queries). In other

words, each node is connected to some nodes that need its result for further processing. When a new result is obtained, the node not only returns it to its user, but also forwards or *pushes* the result to those connected nodes. Similarly, an overlay network is needed to maintain the connections between the nodes based on relationships of nodes' queries.

Both query indexing and result forwarding need the support of an overlay network. However, they have different requirements. Query indexing should be accurate, while the latter should make the system efficient. We introduce our preliminary efforts on this issue in the following.

Sonnet: a subscription overlay network

Sonnet is initially invented for subscription of XML feeds using XPath queries over a P2P network [14]. Each node may publish information in XML format to the system. A node can also subscribe to some information by registering an XPath query in the system. XML documents that match a query would be sent to the querying node automatically by the system.

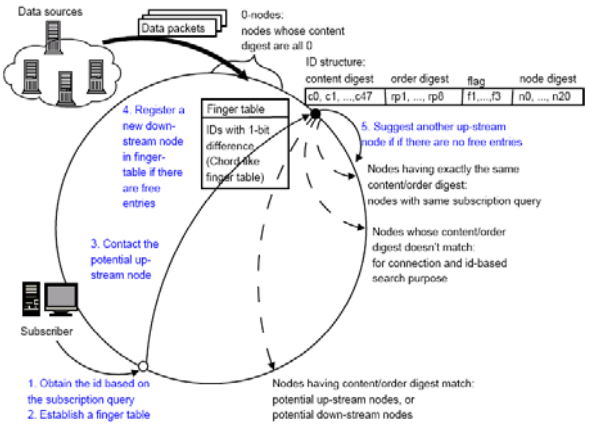


Fig. 3. The network structure of Sonnet.

The structure of the Sonnet network is shown in Fig. 3., while the node architecture is shown in Fig. 4.. Each Sonnet node is also organized in a multi-tier structure. The bottom tier is the Chord [15] overlay network. There is a routing/optimization module that maps the XML document to a 128-bit identifier that forms the head of the XML packet. The mapping preserves the information of both structure and content of the XML documents with little information lost. The head can be used as a key for routing, just as it is conducted in Chord [15]. Thus, the data publishing process is implemented using packet head generating.

XPath queries are also mapped to the 128-bit identifier space. Again, the structure and content of a query is preserved in the identifier. A node with a query is assigned the identifier of that query as the node identifier. Thus, the node identifier assignment is essentially the query register process.

Since the mapping from the XML documents and queries to the identifier is carefully designed, it guarantees that if the identifier matches with an entry in the routing table, i.e. the connection maintained by the node, the packet has high probability to be the answer to the query from that connected node. Therefore, the packet should be forwarded to that node. In other words, it is guaranteed for the overlay network to preserve the relationships among queries. If the packet is transmitted following the links in the routing tables, it will be sent to all nodes that querying it. This fully decentralized scheme efficiently manages the queries in the system. The query processing process is elegantly integrated into packet routing.

Furthermore, the mapping from the XML documents to the identifier is not fixed. It could be adjusted based on the quality of service (QoS) and workload by the nodes. When a node suffers bad QoS, it actively takes over some packet forwarding task from the overloaded nodes. When a node is overloaded, it shares its load to the nodes that it forwards packets to. Intuitively, the system is fair. The node having higher QoS requirements need to take more burdens to serve the system. Therefore, the workload is dynamically balanced. This is a fully decentralized process. Each node's optimization module conducts the balancing based on its local information. The relationships between queries are still preserved by the overlay network after balancing. Thus, queries are indexed by mapping queries to identifiers in the overlay network, while the efficiency is achieved by dynamic optimization over the overlay network. Finally, the partial results are filtered by a local XML query processing engine to generate the accurate final result. Details of the implementation of Sonnet can be found in our technical report [14].

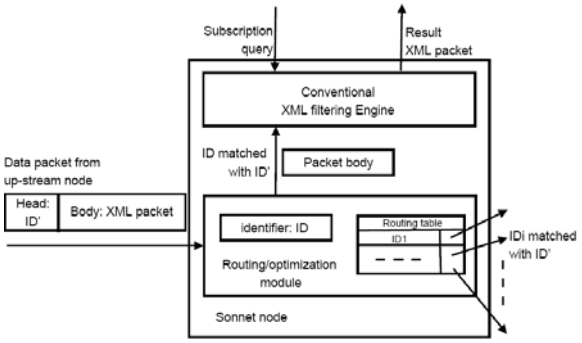


Fig. 4. The architecture of a sonnet node.

In the design of Sonnet, the Chord overlay network is utilized as a basic service. We did not pollute the overlay with any additional operators. So that Sonnet is flexible to be ported to any systems, which are based on Chord. However, the system is still efficient in terms of query processing and network management. This is achieved via in-network processing. Full power of the distributed nodes is utilized.

Other efforts on P2P push-based query processing

Though in the P2P context, the push-based query processing is not as widely studied as pull-based query processing, there are still a few systems developed, including Scribe developed on Pastry overlay, and Multi-CAN developed on CAN overlay [16,17,18,19]. These two systems are both developed over structured P2P networks. However, their designs emphasize more on networking issues than query processing. Only key-based subscription is supported.

Another interesting system is ONYX, a distributed

XML filtering system for large-scale networks, which is similar to Sonnet in terms of functions [20]. It emphasizes on the efficiency of filtering. Different from Sonnet, it does not fully utilize the routing mechanics provided by the overlay network.

5. Conclusions and Discussion

Query processing over P2P systems is a hot research topic. Though there is no killer application like file-sharing or instant messaging found for P2P query processing till now, it is foreseen that P2P-based query processor would be useful in a lot of applications such as financial systems, bioinformatics, and large-scale collaborative systems [6].

In this paper, some basic yet important issues about design and development of an efficient P2P-based query-processing engine are discussed. Two basic approaches, pull-based and push-based methods, are presented. Furthermore, our experiences on developing two representative prototyping systems for each of the two approaches are summarized.

There are some important problems that are not fully solved, including the consistency preserving in a large-scale network, schema management and integration, and trust and security related issues. They are as important as the technologies introduced in this paper for developing a complete query processing system in Internet-scale network, so they all worth further studying.

6. REFERENCES

[1] R. Huebsch, J. M. Hellerstein, et al, Querying the Internet with PIER. VLDB 2003.

[2] S. Abiteboul, O. Benjelloun, et al, Roger Weber: Active XML: Peer-to-Peer Data and Web Services Integration. VLDB 2002:

[3] O. D. Sahin, A. Gupta, et al, Amr El Abbadi: A Peer-to-peer Framework for Caching Range Queries. ICDE 2004.

[4] W.S. Ng, B.C. Ooi et al. BestPeer: A Self-Configurable Peer-to-Peer System (Poster). The 18th International Conference on Data Engineering 2002.

[5] Y. Ren, C.F Sha, et al. Explore the “Small World Phenomena” in a Pure P2P Information Sharing System. In Proceedings of the third IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003).

[6] W.S. Ng, B.C Ooi, et al. PeerDB: A P2P-based System for Distributed Data Sharing. The 19th International Conference on Data Engineering 2003.

[7] W.N. Qian, B. Du, et al. Design and Implementation of a Peer-to-Peer Text Retrieval System. Technical Report, Department of Computer Science and Engineering, Fudan University, 2006.

[8] Z.G. Lu, B. Ling, et al. A Distributed Ranking Strategy in Peer-to-Peer Based Information Retrieval Systems. In Proceedings of the 6th Asia Pacific Web Conference (APWeb'2004).

[9] W.Y. Cai, S.G. Zhou, et al. C2: A New Overlay Network Based on CAN and Chord. International Journal of High Performance Computing and Networking. 2005.

[10] S. G. Zhou, Z. Zhang, et al. SIPPER: Selecting Informative Peers in Structured P2P Environment for Content-based Retrieval. In Proceedings of the 22nd International Conference on Data Engineering (ICDE'2006).

[11] Z. Zhang, S. G. Zhou, et al. KEYNOTE: Keyword Search Using Nodes Selection for Text Retrieval on DHT-based P2P Networks. In Proceedings of the 11th International Conference on Database Systems for Advanced Applications (DASFAA'2006).

[12] F. M. Cuenca-Acuna, C. Peery, et al. PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. In Proceedings of the 12th International Symposium on High Performance Distributed Computing (HPDC) 2003.

[13] C. Q. Tang, Z. C. Xu, Sandhya Dwarkadas: Peer-to-peer information retrieval using self-organizing semantic overlay networks. SIGCOMM 2003.

[14] W.N.Qian, L.H.Xu, et al. Sonnet: Subscription using path queries over structured overlay networks. Technical Report, Department of Computer Science and Engineering, Fudan University. 2006

[15] I. Stoica, R.t Morris, et al, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOMM 2001.

[16] A. Rowstron, A.M. Kermarrec, et al, "SCRIBE: The design of a large-scale event notification infrastructure", NGC2001.

[17] A. Rowstron, P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November 2001.

[18] S. Ratnasamy, M. Handley, et al, Scott Shenker: Application-Level Multicast Using Content-Addressable Networks. Networked Group Communication 2001.

[19] S. Ratnasamy, P. Francis, et al, Scott Shenker: A scalable content-addressable network. SIGCOMM 2001: 161-172.

[20] Y.L. Diao, S. Rizvi, et al. Towards an Internet-Scale XML Dissemination Service. In Proceedings of VLDB2004, August 2004.

An Efficient Clustering Algorithm Based on Quantum-Behaved Particle Swarm Optimization

Xingye Zhang
School of Textiles & Clothing, Southern Yangtze University
Wuxi, Jiangsu, China
Email: zxy1995@163.com
Wenbo Xu
School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, China
Email: xwb@sytu.edu.cn

ABSTRACT

K-Means clustering is one of the widely used clustering techniques, however the major drawback of it is that it often gets stuck at local minima and the result is largely dependent on the choice of the initial clustering centers. An efficient clustering algorithm based on Quantum-Behaved Particle Swarm Optimization, called QPSO-clustering, is presented in this article. Three data sets are employed to test the performance of QPSO-clustering. Performance comparison among k-means clustering, PSO-clustering and QPSO-clustering are also provided. The experimental results show that QPSO-clustering provides better performance than PSO-clustering as well as having less parameter to control than PSO-clustering.

Keywords: K-Means clustering, Genetic Algorithms, Particle Swarm Optimization, Quantum-Behaved Particle Swarm Optimization.

1. INTRODUCTION

Clustering is an important unsupervised classification technique used in identifying some inherent structure present in a set of objects. The main purpose of cluster analysis is to classify objects into subsets that have some meaning in the context of a particular problem[1]. Clustering problems arise in many different applications and play a central role in variety of field [2].

One of the most popular and widely used clustering methods is K-means clustering because it is easy to implement and very efficient. K-means clustering can be described as this: Given a set P of n data points in real d-dimensional space R_d and an integer k, the problem is to determine a partition $P_k = \{C_1, C_2, \cdots C_k\}$, to minimize object function $J(P_k) = \sum_{i=1}^{k} \sum_{\text{xl}\in\text{ci}} d(xl, mi)$, where $m_i = \frac{1}{ni} \sum_{xl \in ci} xl$ is the ith cluster center ,$d(x_l, m_i)$ is the mean squared Euclidean distance from x_l to m_i. Thus, the algorithmic task can be stated as an optimization problem. Although K-Means clustering is one of the widely used clustering techniques, the major drawback of it is that it often gets stuck at local minima and the result is largely dependent on the choice of the initial clustering centers.

Evolutionary algorithms have been introduced to cope with the above mentioned problem of k-means clustering. Genetic Algorithms (GAs) were developed based on the Darwinian principle of the 'survival of the fittest' and the natural evolutional processes. One way to use GAs in k-means clustering is to encode *k* cluster centers in each chromosome. Then the population is iteratively refined by selection of chromosomes, application of mutation and crossover operators, re-evaluation of the new population according to the fitness function for a fixed number of generations or till a termination condition is satisfied. Since k-means clustering based on GAs has integrated the simplicity of k-means clustering with the capability of GAs in avoiding local optima, it provides better performance than k-means clustering[3].Particle Swarm Optimization (PSO) is inspired by the social behavior of animals such as bird flocking, fish schooling, and swarm theory. PSO has comparable performance with GAs [4], and it has been utilized to resolve the clustering problem [5].

2. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization was first introduced by Dr. Russell C. Eberhart and Dr.James Kennedy in 1995[6]. In the Standard PSO model, each individual is treated as a volume-less particle in the D-dimensional space, with the position and velocity of *i*th particle represented as $Xi = (x_{i1}, x_{i2}, \cdots x_{id})$ and $V_i = (v_{i1}, v_{i2}, \cdots, v_{id})$ .The particles move according to the following equation:

$$vid = w * vid + c1 * rand() * (pid - xid) + c2 * Rand() * (Pgd - xid) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

Many research has been done to study the performance of PSO and to improve its performance since its origin in 1995. From these empirical studies it can be concluded that the PSO is sensitive to control parameter choices, specifically the inertia weight, acceleration coefficients and velocity clamping. Wrong initialization of these parameters may lead to divergent or cyclic behaviour.

3. QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION

In [7], Jun sun etc al. introduce quantum theory into PSO and produce Quantum-behaved PSO (QPSO) algorithm. The experiment results indicate that the QPSO works better than standard PSO on several benchmark functions as well as having less parameters to control and it is a promising

algorithm. In Quantum-behaved Particle Swarm Optimization (QPSO), the particle moves according to the following equation:

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \ldots\ldots, \frac{1}{M}\sum_{i=1}^{M} P_{id} \right) \quad (3)$$

$$ppid = \varphi * Pid + (1-\varphi) * Pgd, \qquad \varphi = rand() \quad (4)$$

$$Xid = ppid \pm \alpha * |mbestd - Xid| * \ln(1/u),\ u = Rand() \quad (5)$$

where *mbest* is the mean best position among the particles

p_{id}, a stochastic point between p_{id} and p_{gd}, is the local attractor on the *d*th dimension of the *i*th particle, φ is a random number distributed uniformly on [0,1], u is another uniformly-distributed random number on [0,1] and α is a parameter of QPSO that is called Contraction-Expansion Coefficient.

4. CLUSTERING ALGORITHM BASED ON QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION

Clustering Algorithm Based on Quantum-behaved Particle Swarm Optimization (QPSO) Algorithm, called QPSO-clustering is described as follows.

(1) Initialize an array of particles with random positions and velocities inside the problem space.
(2) Determine the mean best position among the particle

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i = \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \ldots\ldots, \frac{1}{M}\sum_{i=1}^{M} P_{id} \right)$$

(3) Evaluate the desired fitness function for each particle and compare with the particle's previous best values: If the current value is less than the previous best value, then set the best value to the current value. That is, if $J(X_i)<J(P_i)$, then $X_i = P_i$.
(4) Determine the current global position minimum among the particle's best positions. That is: $g = \min_{1\le i\le M} J(Pi)$ (M is the population size).
(5) Compare the current global position to the previous global: if the current global position is less than the previous global position; then set the global position to the current global.
(6) For each dimension of the particle, get a stochastic point between p_{id} and p_{gd} :

$$ppid = \varphi * Pid + (1-\varphi) * Pgd, \qquad \varphi = rand()$$

(7) Attain the new position by stochastic equation:

$$Xid = ppid \pm \alpha * |mbestd - Xid| * \ln(1/u),\ u = Rand()$$

(8) Repeat steps (2)-(7) until a stop criterion is satisfied or a pre-specified number of iterations are completed.

4.1 Population initialization

Each particle is a sequence of real numbers representing the *K* cluster centers. For an *D*-dimensional space, the length of a particle is *D*K* words, where the first *D* positions represent the *D* dimensions of the first cluster centre, the next *D* positions represent those of the second cluster center, and so on. The *K* cluster centers encoded in each particle are initialized to *K* randomly chosen points from the data set. This process is repeated for each of the *M* particle in the population, where *M* is the size of the population.

4.2 Fitness computation

4.3

Fitness can be computed as follows:
(1) The clusters are formed according to the centers encoded in the particle under consideration. This is done by assigning each point x_l, l=1, 2…… n, to one of the clusters C_i with centre m_i if $d(x_l, m_i) < d(x_l, m_j)$, $j=1,2\ldots..k$,$i \neq j$.
(2) After the clustering is done, the cluster centers encoded in the particle are replaced by the mean points of the respective clusters.
(3) Fitness function $J = \sum_{i=1}^{k} \sum_{x l \in ci} d(xl, mi)$ is then computed.

The aim of QPSO-clustering is to search for the appropriate cluster centers $m_1, m_2, \cdots, m_k$ which minimize J.

5. EXPERIMENTAL RESULTS

Three data sets including two real-life data sets and one artificial data set are employed to test the performance of QPSO-clustering and provide performance comparison among PSO-clustering and QPSO-clustering.

5.1 Real world data set

Two data sets which are taken from UCI database repository are considered in our experiment.

1) Fisher's iris data set

Fisher's iris data set (n=150, d=4, k=3): consists of 150 data points, 4 features and 3 classes. Table 1., Table. 2. and Table. 3. are for iris. The main parameters setting in PSO-clustering are in the following: number of iterations=500, population size=10, *c1*=2, *c2*=2. As for *w*, we have four different settings. The main parameters setting in QPSO-clustering are as follows: number of iterations =500, population size=10, α have three different settings.

In Table. 1., the results of three different choices of initial clustering centers are shown. From Table 1. we can find that k-means clustering usually stagnates at local optima if the initial clustering centers are not properly chosen. For Table 2., three different initial populations, three different choices of parameter *w* and respective results are shown. It is found that the result is almost same with different initial population. However, results are different with different *w* settings. When *w*=0.9, PSO-clustering gets stuck at local optimum value of 119.9053.The best value of 96.6555 is obtained when *w* is set to 0.6.For Table 3., three different initial population, three different choice of parameter α and respective results are shown. We can find that QPSO-clustering provides best value of 96.6555 while α =0.9 and never get stuck at local optimum value of 119.9053 while α =0.7 and α =0.8

.

Table 1. Results of k-means clustering for iris data set

Different choice of initial clustering centers	Clustering criteria J
1	97.1233
2	97.3462

3	123.8497

Table 2. Results of PSO- clustering with different value of *w* for iris data set

Different initial population	Clustering criteria J		
	w=0.9	*w*=0.8	*w*=0.6
1	119.9053	119.9053	96.6555
2	119.9053	117.0358	96.6583
3	119.9053	118.5261	96.6555

Table 3. Results of QPSO- clustering with different value of *a* for iris data set

Different initial population	Clustering criteria J		
	α =0.7	α =0.8	α =0.9
1	97.2138	96.6556	96.6555
2	97.4263	96.6687	96.6555
3	97.5182	96.6655	96.6555

2) liver disorders data set

Liver disorders data set (n=345, d=6, k=2) has 345 data points characterized by 6 features and two clusters. Both algorithms are run for 50 times and the mean values of clustering criteria J for 50 times are shown in Table 4. and Table. 5. for liver disorders.

PSO-clustering is realized with the following parameters setting: number of iterations =500, population size=30, *c1*=2, *c2*=2 .As for *w*, we have three different settings. From Table. 4., we can find that PSO-clustering provides the mean value 8450.2 and optimal value of 8108.2 most of the times while *w*=0.4, however 5 times get into local optimum value of 10962.For *w*=0.5, 7 times get into local optimum value of 10962. *w*=0.6 16 times get into local optimum value of 10962.We can find that PSO-clustering can't avoid getting into local optimum even in its best performance while *w*=0.4

Table 4. Results of PSO- clustering with different value of *w* for liver disorders data set

Mean value of Clustering criteria J		
w=0.4	*w*=0.5	*w*=0.6
8450.2	8564.3	9192.3

QPSO-clustering is implemented with the following parameters: number of iterations =500, population size=30, α have two different settings. For Table 5., QPSO-clustering attained the best value of 8108.2 in all the runs when α =0.7.It attains this value most of the times when α =0.8 and never gets into local optimum value of 10962.

Table 5. Results of QPSO- clustering with different value of *a* for liver disorders data set

Mean value of Clustering criteria J	
α =0.7	α =0.8
8108.2	8108.6

5.2 Artificial data Data1

This is a three-dimensional data set where the number of clusters is three. It has 60 data points. The range for the three classes as follows:
Class 1: [15, 25]*[15, 25]*[15, 25]
Class 2: [30, 40]*[30, 40]* [30, 40]
Class 3: [40, 50]*[40, 50]* [40, 50]

Both algorithms are run for 50 times and the mean values of clustering criteria J for 50 times are shown in Table 6. and Table 7 for Data1.The main parameters setting in PSO-clustering are in the following: population size=30, *c1*=2, *c2*=2. Three different settings for *w* and two different settings for number of iterations are given in Table 6..It's found that PSO-clustering attains the best value of 285.92 while *w*=0.5 and number of iteration is taken to 200 in our experiments.

Table 6. Results of PSO- clustering with different value of *w* and number of iterations for Data1

Different number of iterations MaxIt	Mean value of Clustering criteria J			
	w=0.4	*w*=0.5	*w*=0.6	*w*=0.7
MaxIt =100	293.90	286.02	291.84	331.16
MaxIt =200	286.34	285.92	285.95	306.28

The main parameters in QPSO-clustering are as follows: population size=30. Three different settings for *a* and two different settings for number of iterations are given in Table 7..The results show that QPSO- clustering attains the best value of 282.16 while α =0.8 and α =0.9.From Table. 6. and Table 7., we can find that both PSO-clustering and QPSO-clustering get better performance with the increasing number of iteration.

Table 7. Results of QPSO- clustering with different value of *a* and number of iterations for Data1

Different number of iterations	Mean value of Clustering criteria J		
	α =0.7	α =0.8	α =0.9
MaxIt =100	282.94	282.27	283.08
MaxIt =200	282.24	282.16	282.16

6. CONCLUTIONS

An efficient k-means clustering algorithm based on Quantum-Behaved Particle Swarm Optimization, called QPSO-clustering, is presented in this article. The aim of QPSO-clustering is to search for the appropriate cluster centers $m_1, m_2, \cdots, m_k$ which minimize fitness function J .J is described as the sum of mean squared Euclidean distance from data point to their respective clustering centers in this paper. Three data sets including two real-life data sets and one artificial data set are employed to test the performance of QPSO-clustering. Performance comparison among k-means clustering, PSO-clustering and QPSO-clustering are also provided. It's worth mentioning that the performance of both PSO-clustering and QPSO-clustering dependent on the choice of parameters. The experimental results show that QPSO-clustering provides better performance than PSO-clustering as well as having less parameter to control than PSO-clustering.

7. REFERENCES

[1] S. Bandyopadhyay, U.Maulik, "An evolutionary technique based on K-Means algorithm for optimal clusteringin R^N", Information Sciences 146(2002), 221–237.

[2] T. Kanungo, D. M. Mount, "A local search approximation algorithm for *k*-means clustering", Computational Geometry 28 (2004) .89–112.

[3] U. Maulik, S. Bandyopadhay, "Genetic algorithm based clustering technique", Pattern Recognition, 2000,33(9):

1455 - 1465.

[4] E. Elbeltagi, T. Hegazy, "Comparison among five

evolutionary-based optimization algorithms", Advanced Engineering Informatics, 19 (2005), 43–53.

[5] S. Paterlini, T. Krink, "Differential evolution and particle swarm optimization in partitional clustering", Computational Statistics & Data Analysis 50 (2006). 1220 – 1247.

[6] Kennedy J, Eberhart R, "Particle swarm optimization," .

Proceedings of the IEEE international conference on neural networks (Perth, Australia), 1942–1948. Piscataway, NJ: IEEE Service Center, 1995.

[7] J.Sun, B.Feng, "Particle Swam Optimization with Particles Having Quantum Behavior", Proceedings of 2004 Congress on Evolutionary Computation. (2004). 325-331

Study on Product Sale Configuration Decision-making Optimization Blending "context knowledge"

Hong Zhang1,2, Zuzhi Shen1, Wei Jiang1，Shuming Li2
1 The College of Management，Zhejiang University, Hangzhou, 310014 P.R..China
2 Zhejiang Economic and Trade Polytechnic, Hangzhou, 310018 P.R..China
(E-mail:zlycpj @sohu.com)

ABSTRACT

Factors influencing decision-making of product sale configuration include not only configuration but also non-configuration factors. Non-configuration factors can be transferred into context knowledge described by half-configuration factors. Blending context knowledge into the building and solving process of sale configuration optimization model can support the product configuration optimization much better.
And in allusion to an material application background the paper introduces it's decision-making optimizing analysis and it's personalization processing, in a way expands the research and application area of knowledge management.

Keywords: context knowledge, sale configuration optimization model, and decision-making optimization.

1. INTRODUCTION

As an important chain of enterprise production and management, realization of product sale configuration optimization is an important measure for enterprises to improve economic efficiency. Decision-making of product sale configuration usually includes following two steps:
Step1: to choose final products (deciding which products will be sold or not);
Step 2: to decide on sale volume according to product types.
In allusion to step 1, both configuration factor of sale unit efficiency and non-configuration factors such as market demand outlook and consumers' satisfaction have to be considered. So it mainly aims at the step 2 when enterprise adopts linear programming and other methods to realize the optimization of product sale[1]. Now product sale volume becomes the decision-making variable.
To support the whole process of product decision-making, it will be more reasonable to build a product sale configuration optimization model that integrates product type choice influenced by non-configuration factors and sale volume decision-making. The above-mentioned factors affecting product sale configuration decision-making can be regarded as a kind of "apperceive of things around". Despres defined apperceive as "context" and pointed out "all knowledge is meaningless without related context" [2]. Thompson also regarded context as an undivided part of apperceive. (Use context to define apperceive and pay attention to it) [3]。
Kakabadse further pointed out context requirement is the question that needs to be explored by future knowledge management and implement[4]。 It enlightens us that it will be meaningful to group above-mentioned non-configuration factors involving in product sale configuration optimization, conclude the state of these factors form "context knowledge", blend the "context knowledge" into the process of building and seeking solutions to the sale configuration optimization model.

2. STUDY ON THE DESCRIPTION OF "CONTEXT KNOWLEDGE" AFFECTING SALE CONFIGURATION DECISION-MAKING

As we mentioned in introduction, two kinds of factors, configuration and non-configuration, usually affect sale configuration decision-making. Configuration factor refers to product unit efficiency which can be quantitatively described, and non-configuration factors are those that can be qualitatively described, such as market outlook, material supply status, consumers' satisfaction on product quality and life period of product.

With regard to the qualitatively description of non-configuration factors' affection, it usually takes the methods of different levels, for instance, "very strong, strong, common, weak, very weak". If people can give them value by marking them according to different levels, we will obtain half configuration description on how much non-configuration factors affect "context". Because division of factor levels and reasonable value usually depend on the experience of markers (geist knowledge), it gives out "status" reflecting the power of every "context" and composes of "context knowledge".

i ——product no.，i=1,2,….,n

"context knowledge" corresponding to product i is described 5 kinds of fact by using term of "expert system". It is marked as "fact j"，j=1,2…5。
Because of dynamic and changeable market, product unit efficiency that is configuration quantitatively described by unit marginal contribution also appears dynamic. It'd better to have half configuration processing after comparing unit efficiency of every product, mark the "status" of value level of final product i as $L_i^{(1)}$， divide the level into three levels: big, middle, small, which respective value is 2， 1， 0， so "context knowledge" corresponding to value of unit efficiency can be described as:

Fact 1： $L_i^{(1)} \in \{2，1，0\}$， $\forall i$

For non-configuration factors, they can be also dealt similarly:

Mark the "status" that defines market outlook of product *i* as $L_i^{(2)}$, divide it into 3 levels: good, common, bad, and the respective value is: 2, 1, 0. So the "context knowledge" corresponding to the level of market outlook can be described as following:

Fact 2: $L_i^{(2)} \in \{2, 1, 0\}$, $\forall i$

Mark the "status" that defines consumers' satisfaction levels at the quality of products *i* as $L_i^{(3)}$, divide it into 3 levels: very satisfactory, satisfactory, and unsatisfactory. And the respective value is: 2, 1, 0. So the "context knowledge" corresponding to the level of consumers' satisfaction can be described as following:

Fact 3: $L_i^{(3)} \in \{2, 1, 0\}$, $\forall i$

Mark the "status" that defines the life cycle of products *i* as $L_i^{(4)}$, divide it into 2 levels shape or mature, just introduction or wear away, and the respective value is: 1, 0. So the "context knowledge" corresponding to the life cycle of product can be described as following:

Fact 4: $L_i^{(4)} \in \{1, 0\}$, $\forall i$

Mark the "status" that defines material supply tension level of products *i* as $L_i^{(5)}$, divide it into 2 levels: sufficient, tension; and the respective value is: 1, 0, So the "context knowledge" corresponding to material supply tension level of product can be described as following:

Fact 5: $L_i^{(5)} \in \{1, 0\}$, $\forall i$

So fact 1 to fact 5 constitutes a "fact database".

$$LL_i = \sum_{j=1}^{5} L_i^{(j)} \quad i=1, 2, \ldots, n$$

Please note that the value of LL_i comprehensively reflects the market competitiveness of product *i* to a certain extent ($0 \leqslant LL_i \leqslant 8$ $i=1, 2, \ldots, n$). So matching the value of LL_i and the value of L_i^j reasonably will give a cut-in point for "context knowledge" to blend into the process of building sale configuration optimization model and seeking solutions.

3.BLEND "CONTEXT KNOWLEDGE " INTO THE PROCESS OF SALE CONFIGURATION OPTIMIZATION MODELING AND IT'S SOLUTION SEEKING.

3.1 Modeling and analysis

When blending "context knowledge" into the building of sale configuration optimization model, we need to set up decision-making variables of final-choice on product type for "state function". Mark it as T_i,

$T_i = T_i (L_i^{(1)}, L_i^{(2)}, L_i^{(3)}, L_i^{(4)}, L_i^{(5)})$, and $T_i \in \{0, 1\}$, i=1, 2, …n. "T_i=1" means that no. *i* product is ready for sale, otherwise, it will not be sold for the time being.

At the same time, it's necessary to set up final product *i* as decision-making variable for sale volume. Mark it as X_i, *i*=1,2, …., n. The configuration factors of enterprise products include semifinished products and salable final-products. Sometimes, some final products have characteristics of both semifinished product and final product. (Please refer to case in this paper). Mark the sale volume of final product (not semifinished product) as X_{ip}, $i_p \in \{1, 2, \ldots n\}$ p=1,2…G

In the model of sale configuration optimization, the total consumed coefficient of final product *i* to raw material *j* is given out according to product's every direct consumed coefficient of production system, C_{ij} (*i*=1,2…n *j*=1,2…m), so is the total consumed coefficient of final product *i* to every semifinished product *k.*, *aik* (*i*=1,2…n, *k*=1,2…H), and value coefficient of every product is further given out according to price cost of product *i* *i*=1,2,n.Therefore, the reference [5] analyzed the calculating methods of these coefficients based on Lieontive's input& output analysis method. Because production system of enterprise usually has characteristics of "hierarchical step". For these hierarchical step production system, the reference [6] presented more convenient calculating methods of these coefficients based on input and output analysis method. They are not spread in this paper because of the limitation of length.

When total amount of enterprise marginal contribution maximum is used as target, CC_j is the maximum material supply of no. *j* material, BB_k & AA_k are the minimum and maximum throughput of no.*K* semifinished product respectively, B_{1P} & B_{2P} are the minimum and maximum throughput of no. i_p final product (not semifinished product), A1i & $A2_i$ are the minimum and maximum sale volume of no. *i* final product. In such circumstance, the sale configuration optimization model can be described by following non-linear mixed planning model(NP):

$$max\, Z = \sum_{i=1}^{n} d_i T_i (L_i^{(1)}, L_i^{(2)}, L_i^{(3)}, L_i^{(4)}, L_i^{(5)}) X_i$$

$$\text{s.t.} \sum_{i=1}^{n} C_{ij} T_i (L_i^{(1)}, L_i^{(2)}, L_i^{(3)}, L_i^{(4)}, L_i^{(5)}) X_i \leq CC_j, \quad j = 1,2,\Lambda, m$$

$$BB_k \leq \sum_{i=1}^{n} A_{ij} T_i (L_i^{(1)}, L_i^{(2)}, L_i^{(3)}, L_i^{(4)}, L_i^{(5)}) X_i \leq AA_k, \quad k = 1,2,\Lambda, H$$

$$T_i (L_i^{(1)}, L_i^{(2)}, L_i^{(3)}, L_i^{(4)}, L_i^{(5)})\, A1_i \leq x_i \leq T_i (L_i^{(1)}, L_i^{(2)}, L_i^{(3)}, L_i^{(4)}, L_i^{(5)})\, A2_i \quad i=1, 2, \ldots, n$$

$$T_{ip} (L_{ip}^{(1)}, L_{ip}^{(2)}, L_{ip}^{(3)}, L_{ip}^{(4)}, L_{ip}^{(5)})\, B1_p \leq x_{ip} \leq T_{ip} (L_{ip}^{(1)}, L_{ip}^{(2)}, L_{ip}^{(3)}, L_{ip}^{(4)}, L_{ip}^{(5)})\, B2_p \quad i_p \in \{1, 2, \ldots n\}\ p=1, 2, \ldots G$$

$$T_i (L_i^{(1)}, L_i^{(2)}, L_i^{(3)}, L_i^{(4)}, L_i^{(5)}) \in \{0, 1\} \quad i=1, 2, \ldots, n$$

$L_i^{(j)} \in \{2, 1, 0\}$ j=1,2,3 $\forall i$

$L_i^{(j)} \in \{1, 0\}$ j=4,5 $\forall i$

Then mark as follows:

$$CQ_j = \sum_{i=1}^{n} C_{ij}\ T_i\ (L_i^{(1)},\ L_i^{(2)},\ L_i^{(3)},\ L_i^{(4)},\ L_i^{(5)})\ x_i \qquad j = 1,2,\Lambda\ ,m \quad (1)$$

$$AQ_k = \sum_{i=1}^{n} a_{ij}\ T_i\ (L_i^{(1)},\ L_i^{(2)},\ L_i^{(3)},\ L_i^{(4)},\ L_i^{(5)})\ x_i \qquad k = 1,2,\Lambda\ ,H \quad (2)$$

CQ_j and AQ_k refer to the raw material consuming volume of no. j material and the output volume of no. k semi product respectively, both of them fulfill the optimal sale configuration, obviously X_i refers to output volume of no. i final product. So (NP) is actual an enterprise's input & output optimization model using product sale configuration optimization as core and blending "context knowledge". Enterprise has to consider corresponding stock when making kinds of plans. However, the best solution of NP, and the value of CQ_j and AQ_k calculated based on the best solution submit powerful support for enterprise to reasonably and effectively make sale plan, production plan and purchase plan.

3.2 Solution seeking of model

Please notice, in the model of NP, once the value of T_i (i=1,2…n) is ensured, NP changes into LP. Because the solution seeking linear planning is already matured, the key is how to use "context knowledge" described by "fact j" (j=1, 2, …5) to define the value of T_i。

Build following "rule" to describe the standard of ensuring T_i equals 1:

Rule 1: IF $(\prod_{j=1}^{5} L_i^{(j)} \neq 0) \cap LL_i \geq 7$ THEN T_i =1

Rule 2: IF $(\prod_{j=1}^{5} L_i^{(j)} \neq 0) \cap LL_i \geq 6$ THEN T_i =1

Rule 3: IF $(\prod_{j=1}^{5} L_i^{(j)} \neq 0) \cap LL_i \geq 5) \cup (\prod_{j=1}^{3} L_i^{(j)} \neq 0) \cap LL_i \geq 6)$ THEN T_i =1

Rule 4: IF $(\prod_{j=1}^{3} L_i^{(j)} \neq 0) \cap LL_i \geq 5)$ THEN T_i =1

Rule 5: IF $(\prod_{j=1}^{3} L_i^{(j)} \neq 0) \cap LL_i \geq 4)$ THEN T_i =1

It is not hard to find out that above-mentioned rules compose "rule database", and the condition of every rule is rigor degree based on every 'fact" matching when the decision on product sale is made。For "fact h" (h=1,2…5), the lower of h, the more rigor of respective condition.

Mark "state value" of $L_i^{(j)}$ (j=1,2,…,5 i=1,2,…,n) manually, and give the 0 value of T_i (i=1,2,….,n). Through following consequence mechanism, it can form linear plan model (LP_h) respective to rule h.

After taking action for the solution of these 5 linear planning models each, we get five kinds of product sale configuration optimization results ready for choice aiming at different "rigor extent" circs status. Decision-maker can compare these five projects and choose one he prefers. It not only supports making the basic project of production and management plan, but also needs to base on the basic project , in the meantime, it realizes the more effective management direction through sensibility analysis and submits powerful support to decision-making optimization of enterprise production and management through in-depth internal analysis.

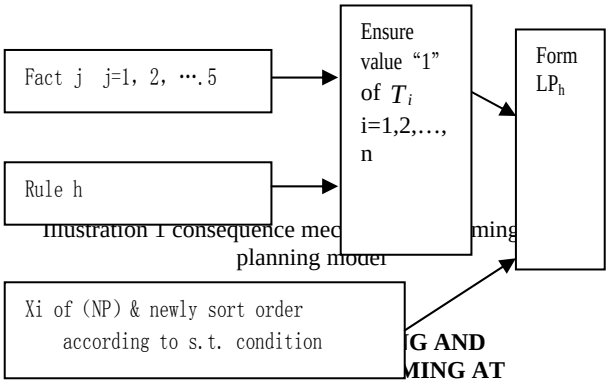


Illustration 1 consequence mec ming planning model

G AND MING AT CONCRETE APPLICATION BACKGROUND

Because different concrete application background usually has "individual" features, it's necessary to personally process the sale configuration optimization model into which blends above-mentioned "context knowledge" during the process of application.

4.1 Personalization processing on a certain case

Showing as illustration 2, production system of a polyamide fibre factory has the feature of "product chain". Obviously, product 1 to product 4 has both characteristics of semifinished product and final product, and it has byproduct during the production. Because material supply is sufficient in the factory, product life cycle is mature, and consumers are very satisfied or satisfied with the quality of its products, fact 4 and fact 5 are deleted from "fact database", while fact 3 can be described as:

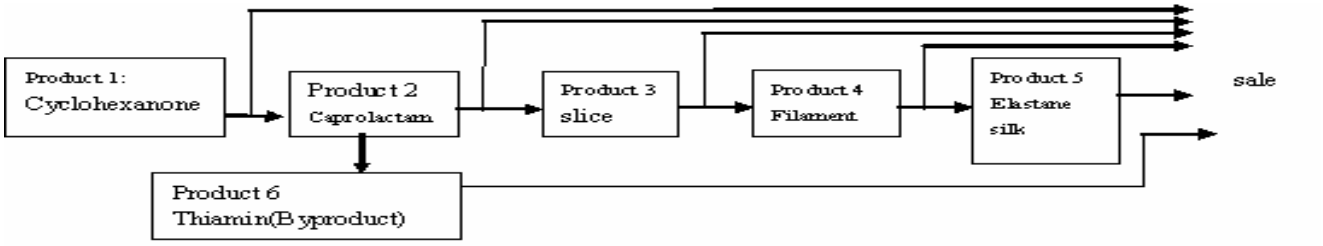


Illustration 2: "product chain" of production system of a polyamide factory

Fact 3: $L_i^{(3)} \in \{2, 1\}$, $\forall i$

Take the output coefficient between byproduct thiamin and main product Caprolactam as α, so the output volume of byproduct is: $x_6 = \alpha(x_2 + \sum_{i=3}^{5} a_{2i} x_i)$, substitute object function of NP, the object function will be changed into:

$$Z = \sum_{i=1}^{5} d_i' T_i \quad (L_i^{(1)},\ L_i^{(2)},\ L_i^{(3)})\ X_i$$

There into $d_1' = d_1 \qquad d_i' = d_i + \alpha\, a_{2i}\, d_6 \quad i$=2,3,4,5

This problem shows that the sale volume of byproduct is not the decision-making variable.

At the same time, because "fact 5" was deleted, the first restrict term of NP was deleted automatically. 3 rues also personally process the "rule database" of this case respectively:

Rule1: IF $LL_i \geq 5$ THEN T_i =1

Rule 2: IF ($LL_i \geq 4$ I $L_i^{(1)} L_i^{(2)} \neq 0$) THEN T_i =1

Rule 3: IF $L_i^{(1)} L_i^{(2)} \neq 0$ Y ($L_i^{(1)} L_i^{(2)} = 0$ I $LL_i \geq 4$)
THEN T_i =1

4.2 Decision-making analysis during the process of application

In application case, according to rule 1 there are T_i =1 (i=1, 2, 3), T_i =0 (i=4, 5), and mark the LP made from reasoning as LP_1. According to rule 2 and rule 3, there are T_i =1 (i=1, 2, 3, 4), T_5 =0, and mark the LP made from reasoning as LP_2.

It doesn't matter under which rule, the last product of product chain in illustration 2: "Elastane silk's" choice variable is 0. It supports the decision that factory plans to suspend the production of Elastane silk and transfer the production line.

After taking action for the solution of LP_1 & LP_2, the optimized value of LP_1 solution is biggest. Shall does it use the optimized project from LP_1 solution as basic project to support plan of production and management? In order to make the decision on basic project choice more effectively, we made sensibility analysis to optimized project from LP_2. We at first judged loosen variables of which s.t equation were 0, then enlarged or lessened constant items by the same standard, and finally optimized one by one so as to understand the most sensitive factors to the increment of efficiency. According to above-mentioned sensibility analysis, it is the most sensitive to the increment of factory's benefit if lessen the value of lower restrict of "Filament" market. Considering that the demand at Filament market fluctuates so much and the unit marginal contribution is so small, it is decided finally to suspend the production of Filament based on basic project of optimized project from LP_1。

In the best solution of LP_1, product "slice" that has the most unit marginal contribution using the lower limit of market restriction for its best solution, while product Caprolactam that has middle marginal contribution using the upper limit of market restriction as its best solution. After analyzing the causes, we find that all products of the factory directly or indirectly consumed its main product: Cyclohexanone. But the output volume of Cyclohexanone is limited (only 6500ton/year). Because material consumed efficient is also one of indexes to scale product unit efficiency [7], it enlightens us to analyze every product's material " Cyclohexanone" unit consumed efficiency. For example, in illustration 2, product i ($i \in \{1, 2, 3, 4\}$), unit using efficiency of material "product 1" can be described as: dd_i (i=1, 2, 3, 4):

$$dd_i = d_i' / a_{1i} \qquad (i=1, 2, 3, 4)\ \ a_{11}=1$$

After calculating, dd_2 is the highest. (the material usage of Caprolactam), It approves that the value of best solution is reasonable. It further enlightens us that it is better to take optimized project based on LP_2 as basic project once dd_4 is the highest, it also provides a gist for the decision of resuming the production of "Filament".

Lowering down the single consumption in production process is related to tapping enterprise internal potential of cost cutting and economic-efficiency-improving. So it is meaningful to analyze the lowering of which single process material consumption is the most sensitive to improve enterprise economic efficiency. Through analysis of sensibility, it is found that lowering the consumption of Cyclohexanone and Caprolactam is most sensitive to the increment of enterprise's economic efficiency. Aiming at the difference between single consumption and recorded highest consumption, sensibility analysis was carried out based on optimized model, when Caprolactam consuming Cyclohexanone and Cyclohexanone consuming benzene reaches to the highest recorded level, the increment of enterprise economic efficiency becomes very sensitive, and

the total increment of enterprise economic efficiency reaches up to 4.154 million yuan, it gives a direction for enterprise to tap potential and improve benefit.

5. CONCLUSION

The research methods and thoughts are also successful applied to the study of "Management method optimization of petrifaction enterprise" and "the demanding forecast of crude oil based on petrifaction stock product", and other specific problems. [8,9]。But its application background and this paper's background is aiming at flow processed enterprises led by stock production. Studies on context knowledge are seldom applied in the dispersed manufacturing enterprises led by order production. It is mainly limited to integrated context share and reuse platform construction[10]. Further study and application of blending "context knowledge" into product sale configuration decision-making optimization is also a meaningful research subject.

7. REFERENCES

[1] Z.Z.Shen, D.L.Xie, et al, China's economic reform promotes operational research applications in the processing industry, International transactions in operational research VOL.8, No. 5 September 2001：560-570

[2] C. Despres, D. Chauval, "A thematic analysis of the thinking in knowledge management [A]", in Despres C, Chauvel D.The Present and the Promise of Knowledge Management [M], ButtereorthHeinemann, Boston, MA, 2000.55-86

[3] M.P. A. Thompson, G. Walsham, "Placing Knowledge Management in Context [J]". The Journal of Management Studies, 2004,41(5): 725

[4] N. K. Kakabadse, A. Kakabadse, et al, Reviewing the knowledge management literature: Towards a taxonomy [J] Journal of Knowledge Management, 2003,17(4): 75-91

[5] A. Abeker, A. Bernardi, et al, "Toward a technology for organizational memories [J]". Intelligent Systems, IEEE [see also IEEE Expert], 1998,13(3): 40-48

[6] A. Abecker, G. Mentzas, et al, "Business-process oriented delivery of Knowledge through domain ontologies [A]". Database and Expert System Applications, 2001. Proceedings. 12th International Workshop on[C], 3~7 Sept.2001.442-446

[7] H.Zhang, Z.Z.Shen, A study on economic benefits analysis of enterprises based on input-output analysis， Donghua University Transactions, Natural science edition march, 2004.3：48-51

[8] H.Li, Z.Z.Shen, et al, Research on optimization Model of production scheduling in a refining and chemical company, Operation Research Transactions, Vol.4 No.2：54-60

[9] G. D. Abowd, E. D. Mynatt, "Charting paste, present and future research on ubiquitous computing [J]". ACM Transactions on Computer-Human Interaction, 2000,7(1): 29-58

[10] X.W.Pan, X.J.Gu, et al, Knowledge Management model of integrated context. Computer integrated manufacturing system，2005.11：65-71

Specifying a Distributed Database Arranged as Ring with the Process Algebra μCRL

Minglong Qi, Qingping Guo, Luo Zhong
School of Computer Science & Technology, Wuhan University of Technology,
Ma Fang Shan Campus, 430070 Wuhan
Email: mlki01@sohu.com Tel: (0086) (27) 87292452

ABSTRACT

In this paper, we have given a specification of a distributed algorithm using the process algebra μCRL. The algorithm is about how preserving the data replication consistency in a distributed database arranged as ring, where each node is attributed a unique integer as priority. An update emitted by a node with greater priority cancels the one emitted by a node with less priority when propagating on this last, while an update originated from a node with less priority will be stopped at a node with greater priority. This strategy has been proven valid by Roscoe et al. The algorithm that we have specified by mean of the process algebra is simplified: each node is supposed to update only one slot. We have given formal definitions of both the process running at a node and the global behavior of the system in the process algebra, and verified that the specification is well-formed using the utilities of the μCRL tool set.

Keywords: process algebra, specification, distributed database, μCRL.

1. INTRODUCTION

The main feature of distributed databases is of distributing and keeping multiple copies of some data in different locations, represented normally by nodes of a network. This schema of the distributed databases has known a wide range of applications, varying from multiple copies of a cache memory in shared-variable parallel computers, through networks of workstations sharing some common information, to widely distributed applications such as automated teller machines. The aim of keeping the multiple copies may be for speed of access, for security (i.e., process P does not lose vital data if process Q goes down), or a combination of the two. The distributed databases systems are generally large and complex, where abstraction becomes a necessity for two reasons, first to understand and reason about them and second to develop and validate the concerned distributed algorithms. Thus, we need really a precise mathematical language adapted to describe the distributed systems in general. In this situation, process algebras such as CSP [3], SDL [4], LOTOS [5], etc. have get their places and shown their power. Among them, we have chosen the specification language μCRL (micro Common Representation Language) for specifying a distributed database arranged as ring, intensively studied by Roscoe and his group [1]. Roughly speaking, μCRL is more mathematically precise than the process algebras mentioned so far, and provides a solid basis for the design and construction of tools for analysis and manipulation of distributed systems.

In section 2 we give a brief and informal description of the process algebra μCRL. The distributed algorithm that guarantees the consistency in a distributed database arranged as ring is presented in section 3. In section 4, we describe formally the result of this paper: specification of the distributed algorithm of Roscoe for a ring database with μCRL. The related works and the future improvement are discussed in section 5. And finally we conclude our work in section 6.

2. PROCESS ALGEBRA μCRL

The language μCRL (micro Common Representation Language) [2] has been conceived as a extension of the process algebra ACP [6] with (equational) data types to describe and verify distributed systems. It offers a uniform framework for the specification of data and process. In μCRL, data are specified by equational specifications: one can declare sorts and functions working upon these sorts, and describe the meaning of these functions by equational axioms. While processes are described in normal process algebraic style, where the particular process syntax stems from ACP [6], enriched with data-parametric ingredients: there exist constructs for conditional composition, and for data-parametric choice and communication. As is common in process algebra, infinite processes are specified by means of (finite system of) recursive equation. For example, the Boolean calculus in μCRL is shown below:

sort $Bool$
func $T, F :\rightarrow Bool$
map $\wedge : Bool \times Bool \rightarrow Bool$
$\vee : Bool \times Bool \rightarrow Bool$
$\neg : Bool \rightarrow Bool$
var $x : Bool$
rew $x \wedge T = x$
$x \wedge F = F$
$x \vee F = x$
$x \vee T = T$
$\neg T = F$
$\neg F = T$
$\neg (T = F) = T$
$\neg (x = F) \rightarrow (x = T)$

sort, **func**, **map**, **var**, and **rew** are all the keywords in μCRL, where **sort** is the keyword for declaring new data type, **func** is the one for declaring constructors of the data type, **map** is the one for declaring functions working upon elements of the data type, **var** for declaring variables and **rew** is the keyword for specifying axioms in form of equations.

Another ingredient of the language is actions that are abstract representations of events in the real world being described. All actions should be declared after the keyword **act,** and may be parameterized. In μCRL, there exist two particular actions, δ and τ, representing respectively deadlocks and internal actions. A deadlock expresses that no

action can be performed, while an internal action represents an action that can take place in a system, but that cannot be observed directly.

Processes should be declared with the keyword **proc**. There are three types of composition between processes: sequential composition, alternative composition and parallel composition. The process $p.q$ is of sequential composition behaving as first performs the actions of p, until p terminates, and continues with the actions in q. The process $p+q$ of alternative composition of p and q, behaves like p and q, depending on which of the two performs the first action. The process $p \,||\, q$ is one of parallel composition between p and q, representing the arbitrary interleaving of actions of the processes p and q, equivalent to $p.q+q.p$ if there is no communication between them.

In this paragraph, we give some basic examples:

1)The first examples declares a simple process, X that performs an a action followed by a b action:

act a, b

proc X = a.b

2)The following declaration recursively defines a process that carries out infinitely many a actions:

proc X = a.X

3)The third example uses parameters to define a counter:

act annonce : Nat

proc Count(n : Nat) =
annoce(n).Count(succ(n))

4)The next specification specifies a process that executes zero or more a actions, followed by a b action, followed by as many c actions as there were a actions.

act a, b, c

proc X = P.δ

P = a.P.c + b

3. DISTRIBUTED ALGORITHM FOR A RING DATABASE

Suppose that we have N processes running on the N nodes of a network in form of ring, and each process has its separate memory, and seeks to hold copies of a same database. The database is changed via updates generated by each process. Those updates are firstly executed locally by their emitters, and ultimately inserted into the ring as soon as possible. After being inserted into the ring, they are circulated around the network and executed by the processes when they arrive at each one according to certain algorithm. In this kind of architecture, the key problem will be how to keep the various copies of a piece of data consistent. In [1], Roscoe has proposed and validated a solution that is called Algorithm 1 by the author, and based on the next stipulations:

One update cannot overtake another when circulating on the ring.

The nodes on the ring are assigned arbitrary but unique priorities. Any process can update any location, and each process should be allowed to execute, locally, its own updates immediately. For node N_i assume that its priority is equal to p_i.

Updates will be the assignment of constant values to single slots. Each node N_i has a queue Q_i of updates, executed locally but not yet entered into the ring, and another queue E_i, which will contain a list of all updates, which it has inserted into the ring and is expecting back.

The Algorithm 1 of Roscoe is described as below:

If the node N_i generates an update u, then u is executed locally and u is inserted at the tail of Q_i.

If Q_i is nonempty and there is a space available on the ring, then the head of Q_i is removed and inserted into the ring, and into the tail of E_i.

If an update u returns which was originated by N_i, then it is removed from the ring, and from the head of E_i.

If an update u arrives that originated at some other N_j and such that there is no clashing update (one to the same slot) in E_i, then u is executed locally and any clashing updates in Q_i are deleted. U is passed on round the ring.

If an update u arrives that originated at N_j with $p_j < p_i$, and E_i contains ones that clash with it, then u is neither executed locally, nor is it passed round the ring. We say that it is stopped.

If an update u arrives that originated at N_j with $p_j > p_i$, and E_i contains ones that clash with it, then these are removed from E_i (they are cancelled), u is executed locally and any clashing updates in Q_i are removed. U is passed on round the ring.

In this article, we will adapt a simplified version of the Roscoe Algorithm, which is based on the next assumption: in the system there is only one slot to update (so that all updates clash).

4. SPECIFICATION OF THE ROSCOE ALGORITHM USING μ CRL

In this section, we will specify the Algorithm 1 of Roscoe about ring database computing using the process algebra μCRL. We need the Boolean calculus and the natural numbers *Nat* with addition and subtraction declared in μCRL, and the map $eq : D \times D \to Bool$ for evaluating if two variables of the sort D are equal. We assume that our ring database has N nodes where N is a natural number, and the generic node is denoted as N_i with its left neighbor denoted as N_{i-1} and its right neighbor denoted as N_{i+1}. We suppose equally that the generic node N_i writes (inserts into the ring) updates to its right neighbor N_{i+1}, and reads updates from its left neighbor N_{i-1} via channels linked between them. The communication between two nodes is supposed to be synchronized. The sort of the slot to be updated is supposed to have n elements where n is a natural number, that are respectively D_1, D_2 … D_{n-1}, D_n. The algorithm is described as the parallel composition of the algorithms for the individual nodes in the ring, which are described generically by means of a linear process equation [7].

In this paragraph, we declare the sort *ID* as identifier of each node that is nothing than the number of the node modulo N.

sort *ID*

func $0,1,2,\cdots N-1 :\to ID$

map $id : Nat \to ID$

var $i : Nat$

rew $id(i) = i \ mod \ N$

In the paragraph, the sort *UPDATE* is defined, representing updates generated by each node. We consider that the assignment of values to a slot of the sort D as the

basic constructor of an update.

sort $UPDATE$

func $:=: D \times D \rightarrow UPDATE$

$Empty :\rightarrow UPDATE$

map $:=: UPDATE \times D \rightarrow UPDATE$

$Val : UPDATE \rightarrow D$

var $x : D$

rew $(x := D_1) := D_2 = (x := D_2)$

$Val((x := D_1)) = D_1$

We has overload the operator := which in the first case constructs an update with assignment of value D_1 to slot variable x, and in the second case changes the value of an update from D_1 to D_2. As for the map *Val*, it simply extracts the value of an update.

In this paragraph, we declare and describe the sort *MESSAGE*, representing the complete information of an update circulating in the ring, and where the identifier of the original node from which the update should have been inserted into the ring, the priority of the original node, and the update its-self should be contained in.

sort $MESSAGE$

func $Empty :\rightarrow Message$

map $mess : ID \times Nat \times UPDATE \rightarrow MESSAGE$

$Origin : MESSAGE \rightarrow ID$

$Priority:MESSAGE \rightarrow Nat$

$Update:MESSAGE \rightarrow UPDATE$

var $i:ID,x:D,p:Nat$

rew $Origin(mess(id(i),p,(x:=D_1)))=id(i)$

$Priority(mess(id(i),p,(x:=D_1)))=p$

$Update(mess(id(i),p,(x:=D_1)))=(x:=D_1)$

$Val(Update(mess(id(i),p,(x:=D_1))))=D_1$

The map *mess* and constant *Empty* are constructors of the sort *MESSAGE*. The maps *Origin*, *Priority*, and *Update* are responsible for extracting respectively the identifier, the priority, and the update of the node emitting this update, from a message of the sort *MESSAGE*.

In this paragraph, we will declare actions a node performs:

act $write,read,write\_read : MESSAGE$

comm $write \mid read = write\_read$

The generic node N_i writes a message containing its update to its right neighbor N_{i+1}, and reads a message from its left neighbor N_{i-1}. The two kind of actions should be synchronized at an action *write_read* onto the channel linking the node N_i and its left neighbor N_{i-1} .

In this paragraph, we will define the process running on the node *Ni*.

Definition 4.1 (Processes). Processes *RoscoeAlgo* are described by means of eight parameters:

- *i*: the identifier of the process.
- *p*: the priority of the node where the process is running.
- u_1: the current copy of the slot that the process holds in form of an update.
- u_2: the update generated and executed locally by the process.
- u_3: the update inserted into the ring by the process and expected to come back.
- m_1: message read from its left neighbor.
- m_2: message sent to its right neighbor.
- *s*: the sate the process is in.

proc $RescoeAlgo(i : ID, p : Nat, u_1 : UPDATE,$
$u_2 : UPDATE, u_3 : UPDATE, m_1 : MESSAGE,$
$m_2 : MESSAGE, s : Nat) =$
$(a) \sum_{d:D} \tau.RescoeAlgo(i, p, u_1 := d, u_2 := d,$
$Empty, Empty, Empty, 1) \lhd eq(s,0) \wedge eq(u_2, Empty)$
$\wedge eq(u_3, Empty) \wedge \neg eq(u_1, Empty) \rhd \delta +$
$(b)\ write(mess(i,p,u_2)).RescoeAlgo(i, p, u_1, Empty,$
$u_3 := Val(u_2), m_1, mess(i, p, u_2), 2) \lhd eq(s,1) \wedge$
$\neg eq(u_2, Empty) \rhd \delta +$
$(c)\ read(m:MESSAGE).RescoeAlgo(i, p, u_1, u_2, u_3,$
$m_1 := m, m_2, 3) \lhd eq(s,2) \rhd \delta +$
$(d)\ \tau.RescoeAlgo(i, p, u_1, Empty, Empty, m_1, m_2, 0)$
$\lhd eq(s,3) \wedge eq(Origin(m_1), id(i)) \rhd \delta +$
$(e)\ \tau.RescoeAlgo(i, p, u_1, Empty, u_3, m_1, m_2, 3)$
$\lhd eq(s,3) \wedge \neg eq(Origin(m_1), id(i)) \wedge$
$Priority(m_1) < p \rhd \delta +$
$(f)\ write(m_1).RescoeAlgo(i, p,$
$u_1 := Val(Update(m1)), Empty, Empty, m_1, m_2, 0)$
$\lhd eq(s,3) \wedge \neg eq(Origin(m_1), id(i)) \wedge$
$Priority(m_1) > p \rhd \delta$

The process can be in four states, denoted respectively by 0, 1, 2, and 3. If s equals 0, the process is in its initial state. After generating an update and executing it locally (represented by an internal action τ), the process is being in the state 1, as expressed by the equation (a). After inserting its update into the ring, the process is moving to the sate 2, where the update expected back becomes u_3:= *Val* (u_2), as described in the equation (b). When an update arrives at the current node, the process is moving in the state 3, the parameter m_1 is updated to that read from its left neighbor, as described in equation (c). If the arrived update has been emitted by the current process, then the process is replaced in the state 0, as described in the equation (d). If the arrived update has been emitted by a process other than the current process and with less priority, then the update is stopped, the process replaces in the state 3, as described in the equation (e). Finally, If the arrived update has been emitted by a process other than the current process and with a greater priority, then the update expected back of the current process is cancelled, and after executing locally the arrived update and sending it to its right neighbor, the process replaces in the state 0, as described in the equation (f).

In this paragraph, we will describe the global behavior of the ring database, which is the parallel composition of all the processes running on the nodes of the ring. For doing this, we need the sort of table of an arbitrary data sort, for example, *MESSAGETable* representing the sort of table of the sort *MESSAGE* defined so far. We omit the definition of the sort of table of an arbitrary data sort, for more detail; please refer to [8].

Definition 4.2 (Parallel composition of processes).

$$RinSys(N:Nat, \mathbf{p}:NTable, \mathbf{u_1}:UPDATETable, \mathbf{u_2}:UPDATETable, \mathbf{u_3}:UPDATETable, \mathbf{m_1}:MESSAGETable, \mathbf{m_2}:MESSAGETable, \mathbf{s}:NTable) =$$
$$RescoeAlgo(0, \mathbf{p}[0], \mathbf{u_1}[0], \mathbf{u_2}[0], \mathbf{u_3}[0], \mathbf{m_1}[0], \mathbf{m_2}[0], \mathbf{s}[0]) \triangleleft N = 0 \triangleright$$
$$(RescoeAlgo(N-1, \mathbf{p}[N-1], \mathbf{u_1}[N-1], \mathbf{u_2}[N-1], \mathbf{u_3}[N-1], \mathbf{m_1}[N-1], \mathbf{m_2}[N-1], \mathbf{s}[N-1]) \,||$$
$$RinSys(N-2, \mathbf{p}, \mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, \mathbf{m_1}, \mathbf{m_2}, \mathbf{s}))$$

Blom and Groote[12] have developed a collection of tools for analyzing systems of communicating processes described in μCRL. We have translated our specification formula into the program that the utilities of the μCRL toolset know treat and proven that the mathematical specification in μCRL of the Roscoe algorithm in this paper is well formed. Due to the space limited of this extended abstract and the complexity of proof, we do not offer verification of the specification of the ring database algorithm here presented, and that we plan to do in a future work.

5. RELATED WORKS

In [9], J. Fredlund and J. F. Groote have specified and verified a leader election protocol using μCRL. In [10], F. W. Vaandrager has worked out of a case study concerning verification of a distributed summation algorithm …briefly we can mention many works done in this line. As for distributed ring databases, Roscoe has specified the Algorithm 1 of a ring database by means of the process algebra CSP in [11], and no works of specifying the same system using μCRL has been done. For future works, we will verify the specification so far described in this note using the general equality theorem and the *cones and foci* proof method of μCRL [7].

6. CONCLUSION

We have utilized the process algebra μCRL for specifying a distributed ring database. We think that μCRL is a very appropriate mathematical tool for describing and validating distributed systems in general.

7. REFERENCES

[1] A.W. Roscoe, "Maintaining consistency in distributed database", Oxford University Computing Laboratory, technical monograph PRG-87, 1990.

[2] J.F. Groote， A. Ponse, "The syntax and semantics of μCRL", in A. Ponse, C. Verhoef, editors, Algebra of communicating Processes, Workshops in Computing, 26-62, 1994.

[3] C.A.R. Hoare, Communicating Sequential Processes. International Series in Computer Science: Prentice-Hall International, 1985.

[4] Specification and description language (SDL). ITU-T Recommendation Z.100, 1994.

[5] ISO/IEC. Lotos – a formal description technique based on the temporal ordering of observation behavior. International Standard 8807. International Organization for Standardization, - Information Processing Systems – Open Systems Interconnection, Geneva, September 1988.

[6] J.C.M. Baeten， W.P. Weijland, Process Algebra, Cambridge Tracts in Theoretical Computer Sciences 18: Cambridge University Press, 1990.

[7] J.F. Groote, J.Springintveld, "Focus points and convergent process operators. A proof strategy for protocol verification", Technical Report 142, University Utrecht, Department of Philosophy, 1995.

[8] J.F. Groote, "A note on n similar parallel processes", in S. Gnesi, D. Latella, editors, ERCIM Workshop on Formal Methods for Industrial Critical Systems, pp. 65-75. Cesena, Italy, 1997.

[9] L.A. Fredlund, J.F. Groote et al, "Formal verification of a leader election protocol in process algebra", Theoretical Computer Science, 177, pp. 459-486, 1997.

[10] F.W. Vanndrager, "Verification of a distributed summation algorithm", in I. Lee, S.A. Smolka, editors, Proceedings CONCUR95, pages 190-203, LNCS 962, Springer-Verlag, 1995.

[11] A.W. Roscoe, the Theory and Practice of Concurrency, London: Prentice-Hall (Pearson), April 2005.

[12] S.C.C. Blom, W.J. Fokkink et al, "A toolset for analyzing algebraic specifications", in G. Berry, H. Comon, editors, Computer Aided Verification (CAV 2001), volume 2102 of *LNCS*, pp.250-254, 2001.

Study of Testability Knowledge Acquisition Based on Graph Theory Model

GuangYao Lian, KaoLi Huang, JianHui Chen, FengQi Gao
Department 4, Ordnance Engineering College
Shijiazhuang, Hebei, 050003,China
Email: ddgcx@163.com

ABSTRACT

It puts forward a testability knowledge acquisition method based on graph theory to solve the testability knowledge acquisition for complex electronic products. Function-structure models are built from using graph theory principle. From calculating the reachable matrix of system, we can get the significant values of nodes. The node that has large significant value is used to divide up the reachable matrix, and dividing up the matrix can get the best fault isolation nodes. Then, we can calculate the amount of information of every node to ascertain the best fault detect nodes. Finally, the best fault diagnosis policy of system can be got. The technology has important meaning in solving the problem of testability-knowledge acquisition in the moment of testability preliminary design. It can also improve intelligent level of testability design and analysis.

Keywords: Knowledge Acquisition, Knowledge Utilization, Graph Theory Model, Testability design, Fault Diagnosis.

1. INTRODUCTION

Testability is an important design character like reliability, maintainability and usability. A good testability design can improve complete quality and reduce full life cycle cost of products. With application of computer technology in the electronic products, it improves the performance and also increases the complexity of whole system at the same time. The test cost of system would be increased greatly, and full life circle cost would be increased at last. So it is important to study the testability design of products. [1]

But the study speed of testability is very slow because it can't build mathematical models of testability that is the basis of quantitative analysis. Many experts both here and abroad have put forward a lot of works in the field. There are two kinds of models now, which are information flow model (IFM) [2] put forward by ARINC Corporation and multi-signal flow graph (MSFG) [3] put forward by Queltech Corporation. This paper gives a function-structure model based on graph theory, which is used to acquire testability knowledge in the moment of testability preliminary design.

2. THEORY FOUNDATION

The reason of why we select graph method to denote testability model is that the graph model has good designability and visualizability, many mature analysis and operation methods can be used, it is easy to compare with the structure models.

As a kind of knowledge representation method for uncertain knowledge, graph theory model includes two parts that are topology and parameter. As for the topology, graph model is a graph; nodes denote the property of real system; rims denote the probability dependency relationship of nodes. We can use conditional probability density functions to describe probability dependency of nodes, which can bring uncertain problems and uncertain reasoning into uniform theory frame and provide a qualitative and quantitative method for probability dependency relationship description of different properties in the problem circumstance. The graph model can be divided into oriented graph model and non-oriented graph model according to the rims of topology. Reference [4] uses an oriented graph model to describe complex electronic system, and the paper also adopts the method.

Oriented graphs of products can be described as $G=G(V,E)$. Here, set of nodes V describes modules of the system; set of rims E describes the bonding strength of nodes. Because the model only uses in the moment of preliminary design, we can support that the bonding length of different nodes that have associate relation is equal. So the testability knowledge acquired in the paper is a rough knowledge. And at the next moment, we can consider the influence of other factors for example fault pattern, reliability, maintainability, manufacturability and so on, reference [5] has researched the problem detailed

3. PROCESS OF MODEL BUILT

This paper puts forward a function-structure model based on graph theory that is used to acquire testability knowledge in the moment of preliminary design and analysis.

3.1 Structure model

Structure models are the detail information models for products' structure characteristic, which describe the relations between modules data and hierarchical structure of products.

It uses structural framework graph method to describe structure model, for example Fig. 1. To describe the interconnect relation of different modules; it defines the following relations in the graph model:

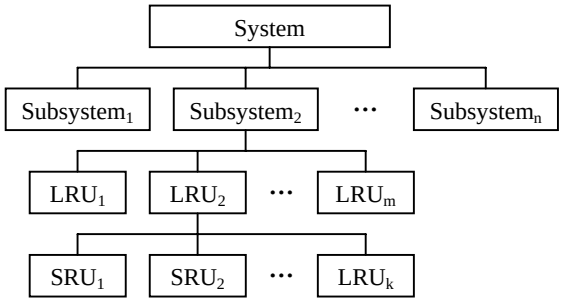


Fig. 1.Structure frame of certain product

Father module: it is the super-stratum module in the graph.

Son module: it is the substrate module in the graph.

In the structure graph, the relation of hiberarchy and modules of system are the principal parts.

3.2 Functional model

Functional models are the detail information models for products' function characteristic, which describe the relation between functional modules and hierarchical structure of products. The structure frame of functional model is same to structure model.

High testability design requests that function should correspond to its structure, which can increase inherent testability design. But many-many relations exist in the actual products. That is to say that many structure modules may realize one function and a structure module may complete many functions. In the process of testability design and analysis, the structure modules are the basis of tasks allocation. So it is important to build the function-structure models of products based on the analysis of function and structure. Fig.2 is the model of function –structure[6].

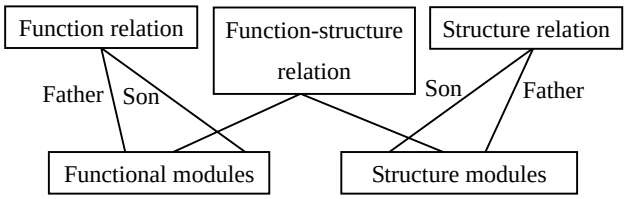


Fig. 2. Function-structure model

3.3 Model built [7]

1) Build up function-structure graph, and get matrixes

According to the relation of signal propagation and structure requests of system, we can get the function-structure graph (*G*) of whole system. The graph is an oriented graph.

It is easy to get the adjacent matrix according to the graph. To get reachable matrix, we can use Warshall arithmetic.

2) Get strongly connected set

After we get reachable matrix, we need determine the strongly connected subsets of *G*. $M=P\odot P^T$ is used to determine the subsets.

The strongly connected subset is made up of nonzero elements of every row in *M*.

3) Decompose work

The nodes of strong component are made up of the elements of strongly connected subset. Then, we can build up a new graph *Gb*.

Step.1: look for all output nodes of *Gb*.

Step.2: a new graph *Gb'* is got after we get rid of the output nodes and the rims connected with them.

Step.3: if *Gb'* has new output nodes, the work jumps to step.2; and if *Gb'* has no output nodes, the work is accomplished.

A hiberarchy function-structure graph will be got from reverting *Gb'* according to the contrary method.

3.4 Knowledge utilization [8]

1) Determination of fault isolation (FI) nodes

Suppose that the reachable matrix of system is $D=[d_{ij}]_{m\times m}$, then the FI significance value of node *j* is $W_{Fij.}$

$$W_{FIj}=\sum_{k=1}^{Z}(N_j^1 N_j^0)_k \tag{1}$$

N^1_j describes the number of *1* in the column vector of node *j*. N^0_j describes the number of *0* in the column vector of node *j*. *Z* describes the number of matrix, $Z\leqslant 2^p$.

After calculating W_{FI}, the large value node will be selected as FI node. The column vector matrix describes as $T_{j.}$

T_j divides the reachable matrix into two parts; they are $D^1{}_p$ and $D^0{}_p$. $D^1{}_p$ is made up of the row vectors correspond to element 1 of T_j. $D^0{}_p$ is made up of the row vectors correspond to element 0 of T_j. At this time, *p=1*， *Z=2.*

Continue to calculate W_{FI} value of $D^1{}_p$ and $D^0{}_p$, and divide up the matrixes, until the matrix has only one row.

Finally, it can be used to guide fault isolation after transforming.

2) Determination of fault detect (FD) nodes

FD node is a key technology to realize good testability design. Here we adopt a principle of information convergence to determine FD node.

The antecedent's set of super stratum node *j*, which is the set of all under layer modules that have fault-transfer relation with node *j*. Information quantity, is:

$$I(j)=\sum_{i=1}^{m} d_{ij} \tag{2}$$

d_{ij} describes the elements of matrix that is made up of the j^{th} column vector.

In many conditions, *I(j)*is a non-null set. We can calculate *I(j)* and judge the information quantity of node *j*. The bigger *I(j)* is, the more information node *j* can provide. We choose the nodes that have more information quantity as FD nodes.

4. EXAMPLE

Fig. 3. is a function-structure graph of certain products.

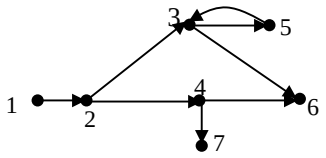


Fig. 3. Function-structure graph of certain product

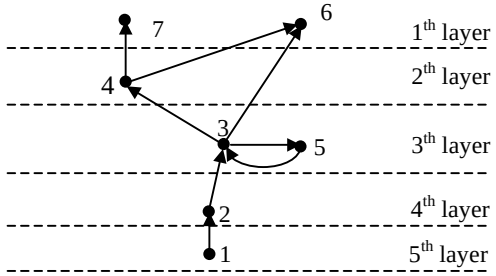


Fig. 4. Hiberarchy function-structure graph

Using the above method, we can get the hiberarchy function-structure graph, Fig. 4.

Because there is a feedback between module 3 and module 5, faults between 3 and 5 can't isolate availably.

So we make 3 and 5 together as module *M*. Then the reachable matrix of the system, is:

$$\begin{array}{c} 1 \\ 2 \\ M \\ 4 \\ 6 \\ 7 \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

1) FD nodes determination

Because *I(6)* equal 5, node 6 will be the first FD node. Then look for super layer node 7, and *I(7)* equal 4, so node 7 is the second FD node. And we can judge the working condition of system from node 6 and node 7.

2) FI nodes determination

After the first detection, matrix of node *7* has only one row, so it isn't necessary to think it over. Then, *Z=1*, *j=,2,…6.*

In the four big W_{FI}, node 7 is selected as first FI node preferentially. We using node 7 to divide the matrix, it is showed by table 1.

Table 1. Matrix division

		Node					
		1	2	M	4	6	7
D^1_1	1	1	1	1	1	1	1
	2	0	1	1	1	1	1
	4	0	0	0	1	1	1
D^0_1	M	0	0	1	0	1	0
	6	0	0	0	0	1	0
7		0	0	0	0	0	1
	W_{FI}	4	6	6	6	0	6
	W_{FI}	2	2	3	0	0	0

From calculating, we can get the next FI node, *M*. Then we can divide matrix by node *M*. Finally, we get the FI node of the system, which are *7, M, 1*.

5. CONCLUSION

Testability knowledge acquisition method based on graph theory, which uses function-structure models of products and get hiberarchy function-structure model by using graph theory. And from dealing with the hiberarchy function-structure graph model, we can get testability knowledge to guide testability design and analysis. By the way, the knowledge acquired by this method is a rough knowledge, and it is effective in the moment of testability preliminary design. But the method has important meaning for improving testability design and testability analysis.

6. REFERENCES

[1] K.L.Huang, "Testability Design and Analysis for Equipments [M]", Weapon Industry Press, 2005,1-2.

[2] Sheppard J, Simpson W., "A Mathematical Model for Integrated Diagnostics", IEEE D&T of Computer, 1992,9(1): 40-54.

[3] Somnath Deb, Krishna R. et al, "Multi-Signal Flow Graphs: A Novel Approach for System Testability Analysis and Fault Diagnosis [J]", IEEE AES systems magazine, 1995,14-25.

[4] X.S.Wen, Y.C.Xu, et al, "Theory and Application of Intelligent Built-in Test [M]", National Defence Industry Press, 2002,51-86.

[5] T.G,Li, "The Research of System-level DFT on Weapon Materiel [D]", Ordnance Engineering College, 2006.

[6] J.Y.Shi, "Research of Choice Method for the Samples in the Testability Demonstration [D]", Beijing University of Aeronautics and Astronautics, 2004,27-39.

[7] G.Y.Lian, K.L.Huang, "Application of Modeling Method Based on Function Hierarchical Decomposition in Knowledge Acquisition [J]", Journal of missile rocket and quidance, 2004,24(2):196-199.

[8] G.Y.Lian, K.L.Huang, "Efficient Algorithm for Testing-Node and Diagnosis Strategies in the Study of Fault-Knowledge Acquisition Based on Simulation[J]", System engineering & electronic technology, 2004,24:1739-1742.

Web Document Classification based on SVM

Qiang Niu, Zhixiao Wang, Dai Chen
School of Computer Science and Technology, China University of Mining and Technology.
XuZhou, JiangSu, 221008, CHINA
Email: niuqiangcumt@126.com

ABSTRACT

With the rapid growth of web information, web document classification has become an important research field for the management of Internet information. Most of the existing methods are based on traditional statistics and they are effective only when the sample size tends to be infinite. They may not work well in practical case with limited samples and it will easily lead to the problem of over-fitting. In order to effectively classify web pages, the paper studies the approach of web document classification in Vector Space Model and feature extraction, and analysis the selection of kernel functions. Based on Support Vector Machine (SVM), a web document classification model and algorithm is proposed. The experiment shows that it can not only improve the training efficiency, but also has good precision.

Keywords: Web document classification, Support vector machines, Kernel function, Feature selection, Statistical learning theory.

1. INTRODUCTION

With the rapid development of World Wide Web, it has been forecasted that these documents with other unstructured data will become the predominant data type stored online. These web documents contain rich textual information, but the rapid growth of the Internet has made it increasingly difficult for users to locate the relevant information on the WWW.

Web document classification, which is the task of assigning natural language texts to predefined categories based on their context, can help organize and search information in the tremendous gallery of web documents. Web document classification presents many challenges and difficulties to the research of Machine Learning. First, it is difficult to capture high-level semantics and abstract concepts of natural languages just from a few key words. Because there are many ways to represent similar concepts and the same word can represent different meanings, for example, semantic analysis, which is a major step in designing a natural language information retrieval system, is not well understood, although there are some techniques that have been successfully applied to limited domains. Second, high dimensionality (thousands of features) and variable length, content and quality are the characteristics of a huge number of documents on the Web. These characters demand place both efficiency and effect demands on classification systems [1].

These challenges and difficulties have led to the development of a number of text classification algorithms, which address these challenges to different degrees. These algorithms include *k*-NN [2], neural network [3], Naïve Bayes [4] and decision tree [5], most of which are either inefficient or too complex.

These classifiers almost have special inherent disadvantages and limitations, respectively. In detail, Nave Bayes classifier needs hypothesis that all features used for classification are submitted to polynomial independence. K-neighbor clustering can be only applied to the case that there is no other class sample in the neighbor of one class. Self-organization neural networks can hold order mapping, but need long training time. For neural network classifier, it needs fewer hypotheses for sample distribution model than Naïve Bayes classifier. So it depends less on special problem, but needs large training samples and more repetition in training. In the case of large samples and high dimension of sample features, the cost of training time is the biggest trouble. If the number of samples is almost in the scale of feature dimension, the samples are very sparse in the distribution space and this must have bad effect on generalization of classifier.

Support vector machines (SVM) work well for classification applications because of their high generalization ability, but on the other hand, most of multi-classifier of SVM is limited on the number of class. It is important to design an appropriate SVM classifier for multi-class classification.

The rest of the paper is organized as follows. The second part introduces the theory of SVM. The third analyses the method to feature extraction. The forth proposes a web document classification model and algorithm, in this part it focuses on design of multi-classifier of SVM and its selection of the kernel functions and parameters. The last part presents some experimental results.

2. SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) foundation was developed by Vapnik and his colleagues [6], which is derived from statistical learning theory and VC-dimension theory. The original SVM is developed to solve binary classification problem.

Theoretically, SVM classification can be traced back to the classical structural risk minimization (SRM) approach, which determines the classification decision function by minimizing the empirical risk, as

$$R = \frac{1}{l}\sum_{i=1}^{l}\left| f(x_i) - y_i \right| \tag{1}$$

Where l and f represent the size of examples and the classification decision function. For SVM, the primary concern is determining an optimal separating hyperplane that gives a tow generalization error. Usually, the classification decision function in the linearly separable problem is represented by

$$f_{w,b} = \mathrm{sgn}(w \cdot x + b) \tag{2}$$

In SVM, the optimal separating hyperplane is determined by giving the largest margin of separation between different classes. This optimal hyperplane bisects the shortest line between the convex hulls of the two classes. The optimal

hyperplane is required to satisfy the following constrained minimization, as

$$\min : \frac{1}{2} w^T w$$
$$y_i (w \cdot x_i + b) \geq 1 \quad (3)$$

For the linearly non-separable case, the minimization problem can be modified to allow misclassified data points. SVM can be applied to multi-class classification by combining SVMs.

3.FEATURE EXTRACTION

In a Web document classification problem, the number of features is quite large, many of which may be irrelevant or redundant. A relevant feature is defined as one removal of which deteriorates the performance or accuracy of the classifier, and an irrelevant or redundant feature is not relevant. These irrelevant features could deteriorate the performance of a classifier that uses all features since irrelevant information is included inside the totality of the features.

Given the peculiarities of web pages differing from plain text documents, we'd make a study on this issue by exploiting how to utilize the HTML structure and how to take hyperlinks into account. Generally, Feature selection is of great importance for hypertext categorization. As is known, during the process of hypertext categorization, a web page is usually presented in VSM (Vector Space Model, though some structural information is taken into consideration), and the native vector space consists of tens of thousands terms for even a moderate sized web pages collection. This is too high for many learning algorithms. In addition, some terms in the space are useless or even harmful to the accuracy of categorization. *So* feature selection is usually to be adopted to reduce the computational complexity and to improve the classification quality. Many methods are available on this issue. We'd select *The Term Frequency-Inverse Document Frequency* weighting method (TF-IDF weighting method, for short) [7]. In this kind of model, Web document space is regarded as vector space composed of series of cross words. Each document can be expressed as one vector of the space, such as $W = \{w_i \mid i \in n\}$, with i is the keyword and w_i be the weight of i in document[8],

$$w_i = \frac{\sum_{i \in S} (w_i \times tf_i) \times \log(N / n_i)}{\sqrt{\sum_j ((\sum_{i \in n} w_i \times tf_j) \times (\log(N / n_j))^2}} \quad (4)$$

tf_i is often defined as the frequency of i in this document，N is the number of training documents, and n_i is the number of documents in which the i-th word appears (so-called Inverse Document Frequency).

In some case it is more important to introduce prior knowledge of samples into selecting features and reducing dimension. For web page classification problem and considering the case of classification based on keyword frequency, the prior knowledge is the statistical information, especially the keyword frequency for each different classes and the keyword concentration degree information in each web page classes. In order to filtrate the redundancy keyword feature, we introduce the concept of word frequency covering rate, as denoting in order to filter useless keyword feature, and the concept of Frequency Covering Rate is referred to in [9].

$$R_{ij} = \frac{tf_{ij}}{\sum_{j=1}^{n} tf_{ij}} \cdot \frac{n_{ij}}{N_j} \quad (5)$$

tf_{ij} is the frequency of the *i*-th keyword in the in the *j*-th document, n_{ij} is the number of the *j*-th class documents in which the *i*-th keyword appears, N_j is the total number of the *j*-th class documents. R_{ij} reflects the correlating degree between the *i*-th keyword and the *j*-th class documents. Bigger R_i is, more precisely it can denote the characteristics of the *j*-th class, and more advantageous it is for the classification. So we can set a threshold *T* to filter keyword features.

$$\overline{R}_i = \max_j \{x_{ij}\} \quad (6)$$

If $\overline{R}_i > T$, then we keep the *i*-th keyword, otherwise we throw it away.

4. MULTISOURCE DATA CLASSIFICATION BASED ON MULTICLASS SVM

4.1 Multiclass SVM Classifier Design

When a linear boundary is inappropriate, SVM can map the input vector into a high dimensional feature space. By defining a non-linear mapping, the SVM constructs an optimal separating hyperplane in this higher dimensional space. Usually non-linear mapping is defined as following function.

$$\varphi : R^n \to Z \quad (7)$$

Feature space *Z* is a Hilbert space and it maps the original vector *x* to a Hilbert space vector $z = \phi(x)$. Also supposing input vector is $x = (x_1, x_2, \ldots, x_n)$,and weight $w_i = \alpha_i y_i$, Multiclass SVM Classifier Design can be illustrated in the following：

Step1: Train datasets

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \quad (8)$$
$$x_i \in R^n, y_i \in \{1,2,\cdots,M\}, i \in 1,2,\cdots,l$$

Step2: Select appropriate kernel function and parameter *C*, conform the optimization

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (K(x_i, y_j) + \frac{1}{C} \delta_{ij}) - \sum_{j=1}^{l} \alpha_j \quad (9)$$

Subject to $\sum_{i=1}^{l} y_i \alpha_i = 0$ and $\alpha_i \geq 0, i = 1,2,\cdots,l$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

So get the optimized result $\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_l^*)$

Step3: Select $\alpha_j^* > 0$, and calculate the threshold

$$b^* = y_j(1-\frac{\alpha_j}{C}) - \sum_{i=1}^{l} y_i \alpha_i^* K(x_i, x_j) \quad (10)$$

Step4: Design decision function

$$f_j(x) = \mathrm{sgn}(g_j(x)) \quad (11)$$

Subject to

$$g_j(x) = \max\left[\sum_{i=1}^{l} \alpha_i^* y_i K(x_i, x_j) + b^*\right] \quad (12)$$

This is process of SVM Classification.

4.2 Kernel function and parameter selection

During the training phase, how to select kernel functions and relevant variables is a complicated problem. The following functions are kernel functions used commonly [10].

- Polynomial Function

$$k(z, z_i) = (z \cdot z_i + 1)^d \quad (13)$$

- Radial Basis Function

$$k(z, z_i) = \exp(-\frac{\|z - z_i\|^2}{\sigma^2}) \quad (14)$$

- Sigmoid Function

$$k(z, z_i) = S[v(z \cdot z_i) + c] \quad (15)$$

Table 1. Accuracy of three kernel functions

kernel function	parameter	SVMs	Accuracy
Polynomial	$d=3$	274	96.0%
Radial Basis	$\sigma^2 = 0.3$	291	95.9%
Sigmoid	$v=0.00781 c=1$	254	95.8%

In the following SVM training and classification experiments for Web document, we combine the three kernel functions with the same parameter, and Table1 shows the result.

The experiments show that the accuracy of three kernel functions is resembled, but because the applicability of Polynomial Function is more extensive, in the following SVM training and classification experiments for Web document we select it as kernel function.

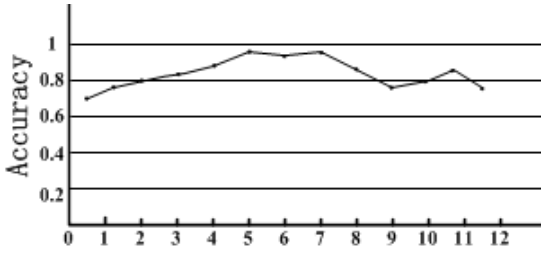


Fig. 1. Accuracy of Polynomial Function

In analysis to relation between *d* and accuracy of classification, the result of this experiment from Fig.1 shows that the accuracy arrive the max when the *d* change from 5 to 7. So in proceeding of Web Document Classification, we select the *d* value in this scope.

5. EXPERIMETNS AND CONCLUSIONS

5.1 Web Document Classification Structure

The procedure of Web document classification sees the Fig. 2.

This proceeding includes two parts: SVM Train and SVM Test. The designed SVM Classifier is trained using a great deal samples in order to regulate the parameter of Decision Function, and it is tested by some other samples to validate efficiency.

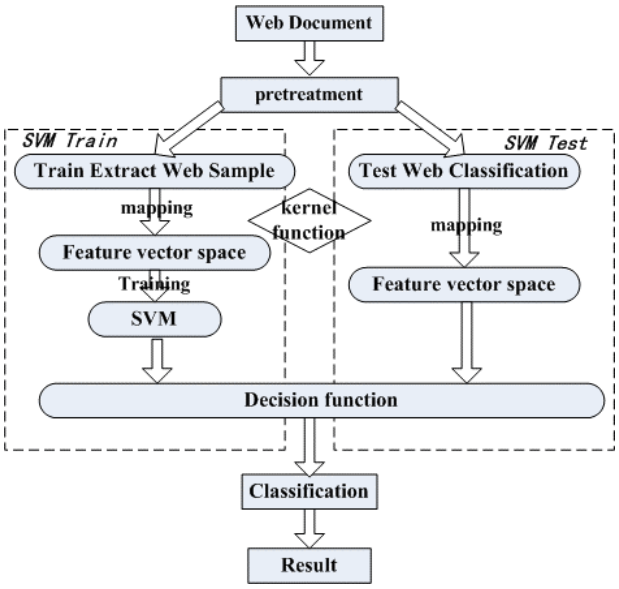


Fig. 2. Procedure of Web document classification

5.2 Datasets Used

Because there is no standard data set for Chinese web page classification, the chosen sample set in this paper is from the CERNET. The whole data is departed into 8 classes with 2090 samples, including 454 samples of mathematics Class, 337 samples of biology class, 315 samples of chemistry Class, 226 samples of physics class, 238 samples of mechanics class, 272 samples of art class, 150 samples of Sport Class, and last 98 samples of literature class. After the word separating process, in the experiment 1403 samples are used in training and 687 samples for testing.

5.3 Performance Measures

The following four quantities are used in several measures to evaluate the performance of a classifier [11].

- *TP* (True Positive): The number of documents correctly classified to that class.
- *TN* (True Negative): The number of documents correctly rejected from that class.
- *FP* (False Positive): The number of documents incorrectly rejected from that class.
- *FN* (False Negative): The number of documents incorrectly classified to that class.

Using these quantities, the performance of the classification is evaluated in terms of Accuracy, Error Rate, Precision, Recall, and *F1* measure defined as follows [8]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

$$\Pr ecision = \frac{TP}{TP + FP} \quad (17)$$

$$\mathrm{Re}\, call = \frac{TP}{TP + FN} \quad (18)$$

$$F1 = \frac{2}{\frac{1}{\Pr ecision} + \frac{1}{\mathrm{Re}\, call}} \quad (19)$$

5.4 Conclusions and future work

From Table 2. we find that if given the right features and classifiers, the Web Document Classification based on SVM algorithm is competitive algorithms in terms of not only the precision and recall but also the subjective assessment. What's more, the algorithm is very fast. So, we think the method is practical.

Table 2. Result of Web Document Classification
(Average of precision, recall and *F1*: 90.11%, 89.38%, 89.7%)

Experiment / Category	Trains	SVMs	Time(s)	Precision	Recall	F1
Mathematics	454	256	36.8	92.2%	89.7%	90.9%
Biology	337	141	27.3	93.1%	90.6%	91.8%
Chemistry	315	152	20.5	89.2%	85.7%	87.4%
Physics	226	248	21.7	84.6%	86.1%	85.3%
Mechanics	238	189	18.4	93.2%	90.3%	91.7%
Art	272	167	9.4	92.5%	90.1%	91.3%
Sport	150	138	6.9	89.1%	89.2%	89.1%
Literature	98	131	7.3	87.0%	93.3%	90.0%

The algorithm for SVM has many advantages compared with conventional machine learning algorithms. The SVM classifier has been used successfully in many fields.

The work of the paper is an attempt for Web Document Classification. However, there are still some problems waiting to be solved. For instance, how to select the most relevant variables? The selection can directly affect the classification result. Otherwise, the speed of SVM training is extremely limited by the number of training vectors. Finding a fast and an effective training algorithm is an important task in actual application of the SVM classifier.

6. ACKNOWLEDGEMENTS

The research has been supported in part of Youth fundamental research program of China University of Mining and Technology (No.OO4490)

7. REFERENCES

[1] Z.H.Deng, S.W.Tang, et al,"Two Odds-Radio-Based Text Classification Algorithms", In Proceedings of 1st International Workshop on Mining for Enhanced Web Search (MEWS 2002, Workshop of 3rd Conference of Web Information Systems Engineering), Singapore, IEEE CS Press ,2002, 223 - 230.

[2] M.Iwayama, T.Tokunaga, "Cluster-based text categorization: a comparison of category search strategies", In Proceedings of the 18th Annul International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95), 1995, 273-281.

[3] H.Ng, W.Goh, et al, "Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization", 20th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SI-GIR '97), 1997,67-73.

[4] Y.Yang, "Expert network: effective and efficient learning from human decisions in text categorization and retrieval", Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94) 1994,11-21.

[5] Freund,Y,Schapire,R,"Adecision-theoretic generalization of on-line learning and an application to boosting", Journal of Computer and System Sciences, 1997,55(1):119-139.

[6] Vapnik.V,The nature of statistical learning theory, New York:Springer,2000.

[7] Salton.G, Michael.J.M, Introduction to modern information retrival, New York: McGraw-hill,1983.

[8] Sebastiani F, "Machine learning in automated text categorization", ACM Computing Surveys, 2002, 34(1): 11-12,32-33.

[9] J.Z.Liang, "Svm Based Chinese Web Page Automatic Classification", Proceedings of the Third International Conference on Machine Learning and Cybernetics,Shanghai,IEEE, 2003,4:2265-2268.

[10] Shawe-Tayler J, Cristianini N, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004.

[11] B.Y. Ricardo, R.N. Berthier, Modern Information Retrieval, Addison-Wesley, 1999.

An Effective Algorithm to Solve Optimal k Value of K-means Algorithm

Taoying Li, Yan Chen
School of Economics and Management, Dalian Maritime University
Dalian, Liaoning, China
Email: ytaoli@126.com, Chenyan_dlmu@163.com

ABSTRACT

In spatial clustering, constructing appropriate validity clustering function is crucial to solve optimal cluster number k. The classical K-means algorithm required the k value to be pre-designated. However, it was difficult to designate precise k value in actual applications, which restricted the extensive application of K-means algorithm. An effective algorithm to solve optimal k value, using distance expense function based on Euclidean distance for verifying validity of optimal cluster number, was given in this paper. At the same time, the upper limit of k value was given and the experience rule was reasonably proved. Above all, computed the inside distance and outside distance of clustering, and computed respective distance expense with different k, and then the k value with minimum distance expense was the optimal cluster number. Finally, the algorithm is applied in practice.

Keywords: Distance Expense, Inside Distance, K-means Algorithm, Outside Distance, Spatial Clustering.

1. INTRODUCTION

Among algorithms of spatial clustering, the most famous and classical partition methods are K-means algorithm, K-center algorithm, and their variations [1]. All above algorithms need to designate the total number k of clusters in advance. However, generally, the k is difficult to pre-appoint, and it is very urgent to research on how to optimize the optimal k. At present, many functions for verifying the validity of clustering are given, but these functions have their limitation and it is difficult to find the optimal k. Thereby, it is necessary to designate a k_{max} (maximum k value), and make $k_{optimization}$ (optimal k value) satisfy: $k_{optimization} \le k_{max}$. How to appoint k_{max} is still undefined, and many researchers use the experiential rules: $k_{max} \le \sqrt{n}$ [2]. The method on how to optimize k value of classical K-means algorithm is researched in this paper.

FKM (Fuzzy k-means) or FCM clustering, brought forward by Bezdek in 1973, is an algorithm that used the degree of membership to judge which cluster each object belongs to. The classical K-means algorithm achieves spatial clustering by means of minimum squared error, and the efficiency of dealing with large data set is better [3]. However, the algorithm requires users to pre-appoint precise k, which restricts the extensive application. Some algorithms are simple and convenient for themselves, but sometimes they are useless in practice. Therefore, this paper brings forward the function of distance expense, and establishes the corresponding model and achieves the algorithm of optimizing k value with minimum distance expense. At the same time, the conditions of solving optimal k value ($k_{optimization}$) and its upper limit k_{max} are given in this paper, and the experience rule $k_{max} \le \sqrt{n}$ is proved rational in theory.

Supported by the National Natural Science Foundation of China (70540005).

2. K-MEANS ALGORITHM

Spatial clustering is an important method to partition spatial data. The spatial objects, according to similarity rule, are partitioned into some subsets, which make the difference of elements in the same subset minimum and that of elements in different subsets maximum. Generally speaking, clustering algorithms is based on various distances, such as Minkowski distance, Manhattan distance, Euclidean distance and Chebyshev distance. Euclidean distance as formula (1) is the most common algorithm.

$$d(X,Y) = \|X - Y\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

X, Y are any two data objects, and X=(x_1, x_2, …, xn), Y=(y_1,y_2,…,y_n).

According to general principles of spatial clustering, the optimal partition should make the comparability maximum and difference minimum in the same cluster, and comparability minimum and difference maximum in different clusters. The distance is used as partition rule. For any object, the distance from it to geometrical center of the cluster that it belongs is smaller than distance from it to geometrical center of other clusters.

The process of K-means algorithm is described as following:

Above all, designate total number k of clusters and choose k objects stochastically. Each object is a seed and stands for the mean or center of a cluster. Each of other objects belongs to the cluster whose center it is nearest. Then compute the new means of each cluster. The process repeats until E in formula (2) converges.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \quad (2)$$

E is squared error sum of all objects, p is an object in space, m_i is the mean of cluster C_i. The visual description of the process is as Fig. 1.

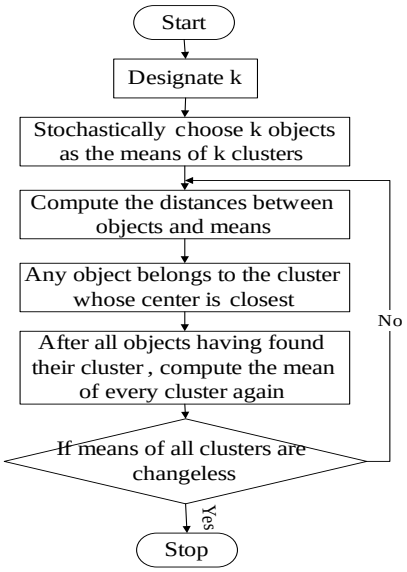


Fig. 1. Process of K-means algorithm

3. DISTANCE EXPENSE FUNCTION AND K VALUE OPTIMIZATION ALGORITHM

3.1 Distance expense function

How to solve optimal k and construct the validity functions of clustering? Different researchers have different answers, some validity functions of clustering are given in reference [4].

1)Segregate coefficient:

$$F(U,k)=\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n}(u_{ij})^2 \tag{3}$$

Supposing Ω is set of all clustering, and the optimal k is given as following:

$$\max_{k}\{\max_{\Omega} F(U,k)\}, k=2,3,\cdots,n-1 \tag{4}$$

2)Segregate entropy:

$$H(U,k)=-\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n}u_{ij}\log(u_{ij}) \tag{5}$$

The optimal k is given as follow:

$$\max_{k}\{\max_{\Omega} H(U,k)\}, k=2,3,\cdots,n-1 \tag{6}$$

3)Effect function of tightness and separability:

$$S(U,k)=\frac{\frac{1}{n}\sum_{i=1}^{k}\sum_{j}^{n}u^{2}{}_{ij}\left|x_j-c_i\right|^2}{\min_{i,j}\left|c_i-c_j\right|^2} \tag{7}$$

The optimal k is given as follow:

$$\max_{k}\{\max_{\Omega} S(U,k)\}, k=2,3,\cdots,n-1 \tag{8}$$

These above functions have their selves' shortcomings, such as nonlinearity characteristic, monotone problem, complexity degree problem, and so on. These disadvantages make it difficult to solve optimal k. The performance of S (U, k) is correspondingly better than others, and it reflects the ratio between the average of distances from objects to the center of clusters that they belong and the distance from objects to their center (mean). The bigger the ratio is, the better the clustering is.

Verifying the rationality of model design is very important. The classical K-means algorithm must satisfy the condition that the k is pre-appointed. Because of better efficiency, the algorithm is always the core of spatial clustering algorithms, but the strict requirement for k restricts its application. This paper constructs the distance expense function and solves the optimal k with minimum distance expense. The algorithm is correspondingly reasonable in practice.

Definition 1: Supposing F={X, R} is clustering space of spatial clustering, X={x_1, x2, …, xn}. If n objects are partitioned into k clusters, the outside distance T is defined as the sum of distances from the mean of each cluster to the mean of all objects.

$$T=\sum_{i=1}^{k}\left|m_i-m\right| \tag{9}$$

In formula (9), m_i is the mean of objects in cluster C_i, m is the mean of all objects, and k is the number of clusters, $m_i=\frac{1}{r_i}\sum_{j=1}^{r_i}x_j, x_j\in C_i, \sum_{i=1}^{k}r_i=n$,

$$m=\frac{1}{n}\sum_{j=1}^{n}x_j$$

Definition 2: Supposing F={X, R} is clustering space of spatial clustering, X={x_1, x2… xn}. If n objects are partitioned into k clusters, the inside distance L is defined as the sum of distances from each object to the mean of cluster that it belongs.

$$L=\sum_{i=1}^{k}\sum_{x_j\in C_i}\left|x_j-m_i\right| \tag{10}$$

In formula (10), x_j is a object called sample, the meanings of m_i, C_i and k are similar to the meanings of them in formula (9).

Definition 3: Supposing F={X, R} is clustering space of spatial clustering, X={x_1, x2,…, xn}. If n objects are partitioned into k clusters, the distance expense function is defined as the sum of inside distance and outside distance multiplying their corresponding weight.

$$\begin{aligned}F(S,k)&=\alpha L+\beta T\\&=\alpha\sum_{i=1}^{k}\sum_{x_j\in C_i}\left|x_j-m_i\right|+\beta\sum_{i=1}^{k}\left|m_i-m\right|\end{aligned} \tag{11}$$

In formula (11), the values of α,β respectively reflect the significance degree of inside distance and outside distance in distance expense, $\alpha,\beta\in(0,2)$ and they satisfies $\alpha+\beta=2$. The meanings of m_i, C_i and k are similar to the meanings of them in formula (9).

The distance expense function is used to verify validity of spatial clustering. The goal is to make distance expense minimum, which is to say that the spatial clustering is optimization when the distance expense minimum. The k value optimization is given as following:

$$\min\{F(S,k)\}, k=1,2,\cdots,n$$

Theorem 1: Supposing F={X, R} is clustering space of spatial clustering, X={x_1, x2,…, xn}. If n objects are partitioned into k clusters, T is outside distance, L is inside distance, when T=L, the k is optimal, which is accord with experience rule: $k_{max}\le\sqrt{n}$.

Prove: Supposing $\overline{L}$ is average distance of distances from each object to the mean of cluster that it belongs, $\overline{L}=L/n$, $\overline{T}=T/k$, considering the clustering

in geometry, the inside structure of each cluster is similar to the whole structure of the clustering, and it satisfies formula (12).

$$\frac{\overline{L}}{L/k} = \frac{\overline{T}}{T} \quad (12)$$

Formula (12) is rational in geometry, and each side of the equation is a ratio between the average distance and total distance of distances from objects to their mean in a cluster.

However, actual clustering space not always has the characteristics of geometry. Generally speaking, spatial clustering must be tight and separable, which is equal to that the partition should make the comparability maximum and difference minimum in the same cluster, and comparability minimum and difference maximum in different clusters. Then it will satisfy:

$$\frac{\overline{L}}{L/k} \le \frac{\overline{T}}{T} \quad (13)$$

If T=L, then $k^2 \le n$, namely $k \le \sqrt{n}$, which is experience rule that many researchers have accepted but haven't proved [5].

3.2 k value optimization of spatial clustering based on distance expense function

Theorem 1 introduces a method to solve optimal total number k of spatial clustering, namely first solves the maximum k, which reduces the range of solutions. The process of this algorithm is described as following:

Step1: according to experience rule, compute the maximum $k \le \sqrt{n}$;

Step2: use K-means algorithm to complete distance of all clustering under $k \le \sqrt{n}$;

Step3: use distance expense function to compute the value of F(S, k) with different k;

Step4: search the minimum F(S, k). If the F(S, k) is the minimum value, then k is optimal, and the algorithm is end.

4. ANALYSIS OF INSTANCE

Supposing there are 9 objects, whose coordinates are shown as Table 1.

Table 1. Coordinates of objects

	p1	p2	p3	p4	p5	p6	p7	p8	p9
x	1	2	3	3	4	2	8	9	10
y	1	2	1	4	5	5	3	2	3

The distribution of these spatial objects is shown as Fig. 2.

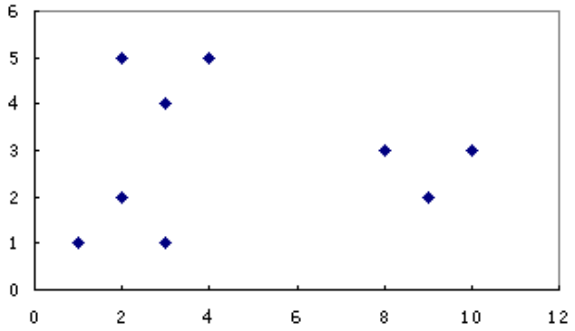


Fig. 2. Distribution of spatial objects

Then the distance expense function is used to solve optimal k. According to experience rule $k \le \sqrt{n}$, $k \le \sqrt{9} = 3$, therefore, the range of k value is reduced as k_1=1,k_2=2,k_3=3, and the step is as following:

Firstly, using K-means algorithm to cluster these 9 objects with k_1=1, k_2=2 and k_3=3, and m_0 is the mean of all objects. In order to simplify the computation, supposing $\alpha = 1, \beta = 1$. The results are respectively shown as Fig. 3, Fig. 4, and Fig. 5.

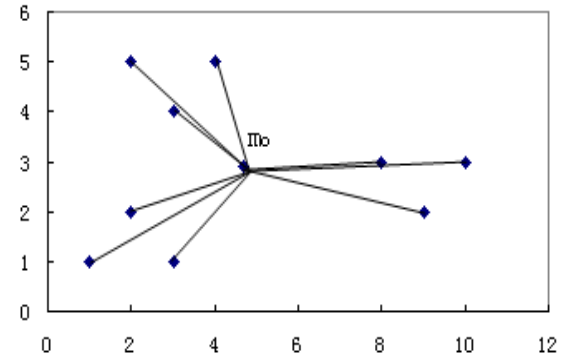


Fig. 3. Distance expense with k=1

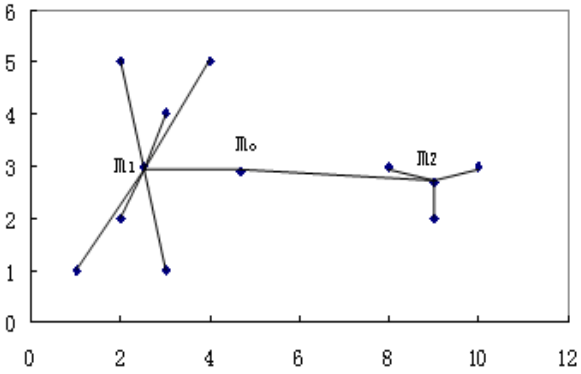


Fig. 4. Distance expense with k=2

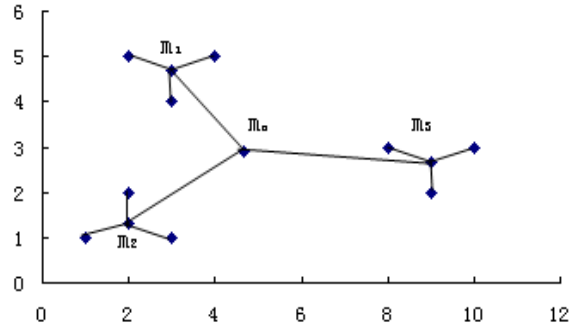


Fig. 5. Distance expense with k=3

Secondly, computing corresponding distance expense, the results are as follow:

F(S, 1) =30.00, F(S, 2) =21.04, F(S, 3) =18.77.

Finally, according to distance expense minimum, k_3=3 is optimal.

Making use of image analysis method is able to quickly find the optimal k. According to formula (11), distance expense function is composed by inside distance, which decreases by degrees, and outside distance, which increases by degrees. According to above paragraphs, the clustering is optimal, namely $k \le \sqrt{n}$, when the T=L. The k may be not the best one but at least the better one. As be shown in Fig. 6, the distance expense is the minimum when L and T intersect, so k=3 is optimal.

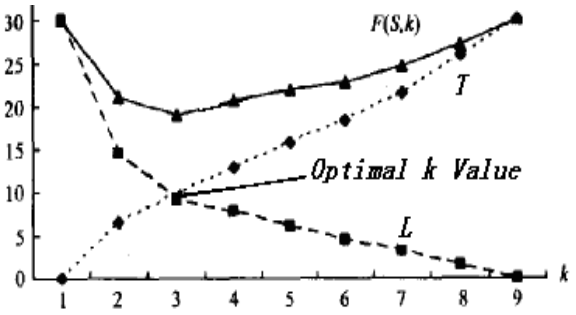


Fig. 6. Trend of distance expense function

5. CONCLUSIONS

Classical K-means algorithm requires the total number k of clusters to be pre-designated, but the precise k value is difficult to pre-appoint in practice, which restricts the wide application of K-means algorithm. This paper gives an algorithm, which can be used to solve the optimal k value and the upper limit of k value. The distance expense function based on Euclidean distance, used for verifying validity of optimal cluster number, is given in this paper. First of all, compute the inside distance and outside distance, then compute the distance expense with different k value, and search the optimal k value. The k value with minimum distance expense is the optimal cluster number. The algorithm is applied in spatial clustering, and the result shows the validity of the algorithm.

6. REFERENCES

[1] S.L.Yang, Y.S.Li, et al, “Optimization Study on k Value of K-means Algorithm”, Systems Engineering-theory & Practice, Institute of China System Engineering, vol.2, Feb 2006, 97-101.

[2] R.M.Ramze, B.P.F.Lelieveldt, et al, “A new cluster Validity indexes for the fuzzy c-mean”, Pattern Recognition Letters, Pattern Recognition Letters, Vol.19, 1998, 239-241.

[3] J.W.Han, Micheline Kamber, Data Mining Concepts and Techniques, Beijing, China Machine Press, 2001.

[4] Z.Z.Shi, Knowledge Discovery, Beijing, Tsinghua University Press, 2002.

[5] J.Yu, Q.S.Cheng, “The range of optimal class number of fuzzy cluster”, Science of China (series E), 2002, 32(2): 274-280.

Data Mining in Web-based Educational System Using An Improved GA

Yunlong Zhou
School of Information Technology, Southern Yangtze University,
No. 1800, Lihudadao Road, Wuxi, Jiangsu 214122, China

Abstract

In this paper we present how to apply an improved Genetic algorithm (GA), which introduce a diversity control method in GA, for data mining of student information obtained in a Web-based Educational Adaptive Hypermedia System. The objective is to obtain interesting association rules so that the teacher can improve the performance of the system. In order to check the proposed algorithm we have used a Web-based Course developed for use by medical students. First, we will describe the proposed methodology, later the specific characteristics of the course and we will explain the information obtained about the students. We will continue on with the implemented Genetic Algorithm (GA) and finally with the rules discovered and the conclusions.

Keywords: Data Mining, Educational System, Genetic Algorithm,

1 Introduction

Nowadays there is a growing trend of web-based technology applied for distance education [1] [2]. But usually, the methodology used to elaborate them is static, that is, when the course elaboration is finished and published on the Internet it is never modified again. The teacher only accesses the student evaluation information obtained from the course to analyze the student's progress. We present, a dynamic elaboration methodology, where the evaluation information is used to modify the course and to improve its performance for better student's learning [3]. The approach is to use a knowledge acquisition method (machine learning and data mining) to discover useful information that might help the teacher to improve the course. We propose an improved Genetic Algorithm for data mining to evaluate the student information obtained from a Web-based Adaptive Hypermedia System. We have used a Web-based Hypermedia Course that was designed to be used by medical student as an example to evaluate our algorithm and to obtain association rules [4] [5] [6]. These rules could then be shown to the teacher in order to help him decide how the course could be modified to obtain best performance.

2 Methodologies

The dynamic construction methodology of Web-based Hypermedia Courses that we propose is recurrent and evolutionary and while the number of students who use the system increases, more information is available to the teacher to improve it. We can distinguish four main steps:

Construction of the course. The teacher builds the Hypermedia Adaptive Course providing information of the domain model, the pedagogic model and the interface module. An authoring tool is usually used to facilitate this task. The remaining information, tutor model and the student model usually is given or acquired by the system itself. Once the teacher and the authoring tool finish the elaboration of the course, then, the full course's content may be published on a web server.

Execution of the course. The students execute the course using a web navigator and in a transparent way the usage information is picked up and stored in the server in a huge database of all the students.

Application of Data Mining. The teacher applies data mining algorithms [8] to the database to obtain important relationships among the data picked up. For this, he uses a graphical data-mining tool.

Improving the course. The teacher using the discovered relationships carries out the modifications that he believes more appropriate to improve the performance of the course. To do it, he again uses the authoring tool. The process of execution, application and improvement can be repeated as many times as the teacher wants to do so, although it is recommendable to have a significant amount of new students' usage information before repeating it.

3 The Web-based Hypermedia Course

We have used the data obtained from the evaluation of a Web-based Hypermedia Course in the study of Rheumatology. The system used to develop the course is an adaptive hypermedia system [6], but the evaluation data used for this paper was obtained from a usability study and there was no attempt to use the result to redesign the Course. As part of this evaluation, the system was used by 30 users, of which 20 were medical and 10 were non-medical users. All the information was stored in a single database in the following tables: USER: String value that represents a system user, in our case they are 30.
PERFORMANCE: Real value that represents the user's performance in the 7 case studies in this application.
AVEP_AH: Real value that represents the average performance of the users in the 7 case studies, adaptive application version.
AVEP_NOAH: Real value that represents the average performance of the users in the 7 case studies, but in the version of the application without adaptation.
CASETIME: Integer value that represents the time that a user takes in visualizing a complete case study.
CASESCORE: Integer value that represents the score that an user has obtained when undertaking a case study. ACCESSTIME: Real value that represents the number of times a user has accessed the application.

CONCEPT: Real value that represents the user's effort spent in the different concepts.
QUESTIONSCORES: Integer value that represents the score obtained by the users in the relating questions to the case studies. The data was preprocessed so that it will be easier to obtain relationship rules from them. This transformation consisted of a discretization, which mapped from continuous values (usually real values) to discrete values (strings that represent values groups) and integer values only needed to be labeled. In this way the modifications made to the tables are as follows: PERFORMANCE, AVEP_AH, AVEP_NOAH, CASETIME, ACCESSTIME and CONCEPT have been discretized to the labels VERYHIGH, HIGH, MEDIUM, LOW and VERYLOW. The values of CASESCORE and QUESTIONSSCORES have been named with the labels SUCCESSFIRST, SUCCESSSECOND, SUCCESSTHIRD and SUCCESSFOUR, which means getting the answer correct at the first attempt, second attempt and so forth. USER does not need modification.

4 An Improved Genetic Algorithm for Data Mining

Some of the main data mining tasks are [8]: classification, clustering, discovery of association rules, etc. We have used an improved Genetic Algorithm to obtain association rules from the user evaluation data. The association rules relate variable values. They are more general than classification rules due to the fact that in association rules any variable may be in the consequent or antecedent part of the rule. The classical problem of discovering association rules is defined as the acquisition of all the association rules between the variables [4]. Genetic algorithms are a paradigm based on the Darwin evolution process, where each individual codifies a solution and evolves to a better individual by means of genetic operators (mutation and crossover). In this paper we use an improved Genetic Algorithm with the diversity controlled for data mining. The improved GA Process we have used consists of the following steps:

4.1 Initialization

Initialization consists of generating a group of initial rules specified by the user (50 - 500 rules). Half of them are generated randomly and the other half starting from the most frequent values in the database. We use a Michigan approach in which each individual (chromosomes) encodes a single rule. The format of the rules we are going to discover is: *IF Variable1 = Value1 (AND Variable2 = Value2) THEN*
VariableX =ValueX
Where:
Variable1, Variable2, VariableX: Are the database's field names. (P0...P6, AVEPH, AVEPNOH, CASETIME0...CASETIME6, CASESCORE0.CASESCORE6, ACCESSTIME0..ACCESSTIME6, C0...C37, MCQ0.. MCQ6).
Value1, Value2, ValueX: Are the possible values of the previous database fields (VERYLOW, LOW, MEDIUM, HIGH, VERYHIGH, SUCCESSFIRST, SUCCESSSECOND, SUCCESSTHIRD, and SUCCESSFOUR).

We use value encoding in which a rule is a linear string of conditions, where each condition is a variable-value pair. The size of the rules is dynamic depend of the number of elements in antecedent and the last element always represents the consequent.

4.2 Evaluation

Evaluation consists of calculating the fitness of the current rules and keeping with the best ones. To calculate the fitness we count the precision of the rule, the number of patterns in the database that fulfill both antecedent and consequent and do not fulfill both antecedent and consequent. That is, we obtain very strong association rules [2] that fulfill [A=a] _ [C=c] and [C≠c] _ [A≠a]. So a rule is very strong if the previous two rules are strong, that is, both rules have greater support and confidence than a minimum values set by the user (0.5-1). Our formula detects both statistical negative dependence and independence between antecedent and consequent.

4.3 Selection

The selection chooses rules from the population to be parents to crossover or mutate. We use rank-based selection that first ranks the population and then every rule receives fitness from its ranking. The worst will have fitness 1, second worst 2, etc. and the best will have fitness N (number of rules in population). Parents are selected according to their fitness. With this method all the rules have a chance to be selected.

4.4 Reproduction

Reproduction consists of creating new rules, mutating and crossing current rules (rules obtained in the previous evolution step). The user sets the crossover and mutation probabilities. A higher crossover rate (50-95%) and a lower mutation rate (0.5-2%) are recommended. Additionally it is good to leave some part of population survive up to next generation. Mutation consists of the creation of a new rule, starting from an older rule where we change a variable or value. We randomly mutate a variable or values in the consequent or antecedent. Crossover consists of making two new rules, starting from the crossing of two existent rules. In crossing the antecedent of a rule is joined to the consequent of the other rule in order to form a new rule and vice versa (the consequent of the first rule is joined to an antecedent of the second). So it is necessary to have two rules to do the crossover.

4.5 Diversity Controlling

After the above genetic operations, the diversity controlling procedure is executed. The diversity of the population is measure by the average Hamming distance among any pair of individuals. If the diversity decline to below a pre-specified low bound (the value is set to 1 in this work), the mutation rate will be set to 0.1 until the diversity return to above the low bound. This diversity control method could enhance the global search ability of the GA and result in a good performance.

4.6 Finalization

Finalization is the number of steps or generations that will be applied to the genetic process. The user chooses this value (10-500 steps). We could also have chosen to stop when a certain number of rules are obtained.

5 Results and Conclusions

We have applied the algorithm to the whole data, only to the medical students and only to the other users. For each case, we have obtained different rules both in the content and in number and fit. For example, one of the rules obtained when using 100 initial rules and 100 steps is:
IF CASESCORE2=SUCCESSFIRT AND CASESCORE4=SUCCESSFIRT THEN CASESCORE1=SUCCESSFIRST (with support = 0.73 and confidence=1).

This can be interpreted as if a user gets the answer for case 2 and case 4 correct, he/she is likely to do well in case 1. The support of a rule gives the importance of a rule and the confidence of a rule gives its predictability power. All the rules discovered are showed to the teacher in order he can obtain conclusion about the course functionality. The teacher has to analyze them and he has to decide what are the best modifications that can improve the performance of the course. Summarizing the main conclusions that we obtained starting from the discovered rules are: We obtained expected relations, for example: between CASESCORE and CONCEPT, and between MCQ and CONCEPT, due to the fact that the questions are about the concepts. We obtained useful relations, for example: between CASESCORE and MCQ. This could be because the questions are the same ones, or they refer to the same concepts, or they have equal difficulty. We obtained strange relations, for example between AVEP_AH and P, it is probable that this relation takes place by chance. And we didn't find any other relation, for example with ACCESSTIME or CASETIME. This could be because user access times were completely random and they did not determine any other variable as it might be expected.

References

[1] B.P. De, H. Wu, et al, "Adaptation Control in Adaptive Hypermedia Systems", International Conference on Adaptive Hypermedia, 2000.
[2] M. Delgado, D. Sánchez, et al, "Mining association rules with improved semantics in medical databases", Artificial Intelligence in Medicine, 2001.
[3] A.A. Freitas, "A survey of evolutionary algorithms for data minig and knowledge discovery", to appear in: Advances in Evolutionary Computation. A Springer-Verlag, 2002.
[4] J. Hipp, U. Güntzer, et al, "Algorithms for Association Rule Mining – A General Survey and Comparison", SIGKDD Explorations, 2000.
[5] M. Ng, W. Hall et al, "History-based Link Annotation for Self- Exploratory Learning in Web-based Hypermedia", 3rd workshop on Adaptive Hypertext and Hypermedia. Arhus, Denmark, 2001.
[7] R.Z. Osmar, "Web Usage Mining for a Better Web-Based Learning Environment", Technical Report. 2001.

Optimization of Query Plan in Data Stream System*

Jinxian Lin
Network Information Center, Fuzhou University
Fuzhou, Fujian 350002, China
Email: jxlin@fzu.edu.cn
And
Zhanping Zhen
College of Mathematics and Computer Science, Fuzhou University
Fuzhou, Fujian 350002, China
Email: zhizhiping@fzu.edu.cn

ABSTRACT

According to the data characteristics, the data arrival rate, the selectivity and the correlation among the query operations in data stream, a core algorithm called CSOQP (Combination Section Optimization Query Plan) is proposed. By the block size of data stream, CSOQP can release the contradiction between the cost of the scheduling processing and speed of the query processing. With a proper parameter, CSOQP can easily cope with the fluctuating problem in selection of query plan and strongly optimize the query plan in space and time.

Keywords: Data Stream, Query Plan, Query Algorithm, Query Optimization, Database.

1. INTRODUCTION

The focus of research on data and information processing has recently shifted towards an emerging type of applications in which the data is streaming from its sources. Such applications include monitoring network traffic, intrusion detection, telecommunications, sensor networks, financial services, and e-business applications.

Some major assumptions made by traditional data management systems do not hold in the context of streaming applications. In these applications, the system has no control over the arrival time of the data. Hence, the adoption of a push model of computation is mandatory. Also, in such applications, monitoring queries can run for a long time (e.g., on the order of days or months) so that they can be assumed for all practical purposes to be running continuously, hence the name continuous queries.

We use the data stream processing model proposed by Stanford University in its STREAM data stream system, but the adaptation of selection of the query plan is not very good for its simple processing in the STREAM. We address the adaptation problem and the optimization problem by proposing the core algorithm called CSOQP (Combination Section Optimization Query Plan). We use the EDDY processing model to route the tuples. Within the EDDY, we can know and control the tuples' direction. In [1], it can route every tuple in special way, and in [2], it analysis the cost model of EDDY. As in [2] show, the cost of EDDY is not very acceptable, we consider to use the block processing instead, according to the size of the block, we can release the contradiction between the cost of the scheduling processing and speed of the query processing.

Supported by FPNSFC A0410011 and FSTSF 2005HZ1011.

2. QUERY PLAN

In data stream system, we use a defining query language to describe query requirement, as CQL in STREAM system. For simplicity, we still regard the CQL as the standard query language. In most data stream system, a query operation usually can be decomposed several original operations. We use an operation tree to present the query requirement, finish the original operations from bottom to top with the tree, and get the result tuples finally. Such as the example following:

$$select\ s.a\ from\ s\ where\ s.a = 1\ and\ s.b = 2\ and\ s.c = 3 \quad (1)$$

$$select\ s.a\ from\ s\ where\ s.a = 1\ and\ s.b = 2\ and\ s.c = 4 \quad (2)$$

We denote s a input stream and $s.a$, $s.b$, $s.c$ three properties of the stream. According to the query Eq. (1) and (2), we can build the query trees. Fig.1 shows the trees.

***Definition* 1**. A query plan is derived from one query, but one query also has the several query plans. If the query has n original operations, it must have $n!$ query plans.

PROOF: A query contains n original operations and we denote them O_1, O_2, $\cdots$, O_n. According to the difference of the form order, we can see the different query plans, so the problem is equal to how to arrange these operations. With the knowledge about order problems in mathematics, we know that, to the n number, the combination number of the order without overlap must be n. So we can proof our point.

In data stream system, we often deal with the tuples in pipelined way, because once an operation is blocked, the system will be blocked and the memory will be exhaustion. Through the pipelined filtered operations, every operation just like a filter not blocked, so we can see that the processing of the tuples actually just is the processing of an array of filtered operations, and system will get the filter result finally. Join operation must be also dealt with in such way; we can find the solution in [3, 4].

Obviously, the original operations run time overhead and memory requirement overhead is change over the infinite data, so it has some complexity to choose the most optimum query plan. For example, as to a simple predication, $s.a$ = '*abcd*', to complete the operation, we must use the prefix-match measure to find the matched tuples quickly. The speed of such operation is quick, but some operations may use heavy cost and much resource to complete them.

For example, we use user-defined procedure, like *domain* (*s.a*) = '*fujian*'. If we do not consider such diversity, system may waste much resource and do few things.

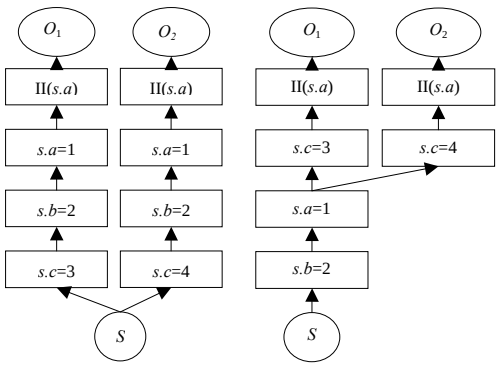


Fig.1 Query trees

Otherwise, the selectivity of the every original operation often is different, and if we do not consider it, it will not be an optimal query plan, because it also seriously affect the processing performance, we can see the CQL query as follow:

$$select\ s.a\ \ from\ s, t\ where\ s.a = t.a\ and\ s.a = 1 \qquad (3)$$

From Eq. (3), there are two different query plans as Fig.2. To the original operations O_1, O_2, O_3, their selectivities are different. We denote their selectivities S_1=0.1, S_2=1.1, S_3=1 respectively, and denote the time cost that is spent processing the each tuple T_1=1*ms*, T_2=10*ms*, T_3=1*ms*, assuming that the time cost of processing the left tuples is equal to the cost of the right tuples, namely, T_2 (*left*)=T_2 (*right*)=10*ms*. The analyses the cost of the several different join algorithms in detail [5].

From Fig.2 (a), if the query plan is implemented like that, and we also know the several parameters as follow, we will know the cost of this query plan.

From Fig.2 (b), we also know the cost as follow.

$$T(3) = T_1 + S_1T_2 + T_2 + T_3S_1S_2 = 12.11ms \qquad (4)$$

$$T(4) = 2T_2 + S_2T_1 + S_1S_2T_3 = 21.21ms \qquad (5)$$

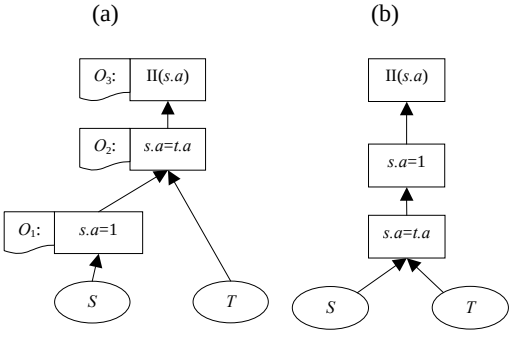


Fig.2 Query plans

For T (3)<T (4), we can infer that the query plan illustrated with Fig.2 (b) is better than Fig.2 (a). Meanwhile, assuming data arrival rate of stream *s* equal to data arrival rate of stream *t*, R_s=R_t=50 tuples per second, we must notice that the costs of the two query plan are T (3*total*) = T (3) *Rs* = 605.5*ms*, T (4*total*) = T (4) *Rs* = 1060.5*ms*.

As show above, T (4*total*)>1*s* dedicates that the number of the arrival tuples is more than the number of the tuples that the system can process. We can store the extra arrival tuples in buffers, when the system is idle, and then process the tuples in the buffer. When the buffers are still not enough, we can use other technologies to cope with the problem, like load shedding [6, 7].

As to query containing *n* original query operations, we introduce an algorithm below called CSOQP Algorithm, which can decide nearly optimal query plan. In data stream, the data arrival rate change and the data characteristics also change over the time, so our algorithm must adapt these factors in a way. To implement the adaptation, our algorithm must deal with the two problems as follow:

- ***Adaptation algorithm run-time overhead***: The adaptation algorithm must know the every operation's selectivity and the time cost of processing each tuple. However, as to the varying data arrival rate and data characteristic, the two factors may change endlessly simultaneity. In section 3, we propose CSOQP algorithm to resolve the problem by just capturing some tuples as sample, and using proper sample technology, we can ease the cost of the CPU and memory in a way.
- ***Adaptation fluctuation***: In some special applications, their original operations' selectivities and their time cost of processing each tuple may change very frequently, so if we do not control the speed of adaptation, system will use much resource to complete the adaptation.

3. THE COST ANALYSIS OF QUERY PLANS

We consider the situation that a query ***Q*** filters the input stream ***I***, and denote the filtered operations F_1, F_2, ⋯, F_n, their selectivities S_1, S_2, ⋯, S_n, and time cost of processing each tuple T_1, T_2, ⋯, T_n. Assuming that the order of the original operations which the query plan what we chose contains is like F_{x1}, F_{x2}, ⋯, F_{xn}. We can infer the time cost of F_{xi}

$$C(F_{xi}) = \begin{cases} \prod_{x=1}^{i-1} S_x T_i & i > 1 \\ 1 & i = 1 \end{cases} \qquad (6)$$

and the total time cost of the query plan is

$$C(n) = \sum_{i=1}^{n} T_i D_i$$

$$D_i = \begin{cases} \prod_{x=1}^{i-1} S_x & i > 1 \\ 1 & i = 1 \end{cases} \qquad (7)$$

***Definition* 2**. If all of the selectivities of the original filtered operations have the same value *S*, and *S*<1, the operations should be sorted ascending by time cost of processing each tuple for getting the optimal time cost.

PROOF: By using the reduction to absurdity, we proof the definition. So, if the conditions above are true, and the operations are not sorted ascending by time cost of processing each tuple, We can find at least one pair of O_i and O_j, which sequence number are *x* and *y* respectively and *x*<*y* that O_i is executed before O_j and T_i>T_j. With this deduction, by reversing the two operations' positions, we can get the other query plan. The time cost of the original query plan is showed follow:

$$C(n) = C(x-1) + \prod_{k=1}^{x-1} S_k T_i + \prod_{k=1}^{x} S_k T(x+1) + \cdots + \prod_{k=1}^{y-1} S_k T_j + C \qquad (8)$$

The time cost of the modified query plan is showed

follow:

$$C'(n) = C(x-1) + \prod_{k=1}^{x-1} S_k T_j + \prod_{k=1}^{x} S_k T(x+1) + \cdots + \prod_{k=1}^{y-1} S_k T_i + C \quad (9)$$

We denote C total time costs of all operations, which sequence numbers, are larger than y. Because $S<1$, $x<y$ and $T_i>T_j$, C' $(n)<C$ (n). It is not inconsistency with our assuming argumentation. So it is proofed.

***Definition* 3**. If the time costs of processing each tuple of the original filtered operations have the same value T, the operations should be sorted ascending by the size of the selectivity of the original filtered operations for getting the optimal time cost.

PROOF: By using the reduction to absurdity, we proof the definition. So, if the conditions above are true, and the operations are not sorted ascending by the size of the selectivity of the original filtered operations, we can find at least one pair of operations O_i and O_j, which sequence number are x and y respectively and $x<y$ that O_i is executed before O_j and $S_i>S_j$. By reversing the two operations' positions, we can get the other query plan. We compare the two query plans by their total time cost. The time cost of the original query plan is showed follow:

$$C(n) = C_1 + C_2 + C_3 + C_4 + C_5 \quad (10)$$

C_1 is the total time cost of the operations before x;

$$C_2 = \prod_{k=1}^{x-1} S_k T;$$

$$C_3 = S_i T \sum_{m=x+1}^{y-1} \prod_{k=1}^{x-1} S_k (1 + \prod_{k=x+1}^{m} S_k);$$

$$C_4 = \prod_{k=1}^{y-1} S_k T;$$

C_5 is the total time cost of the operations after y.

The time cost of the modified query plan is

$$C(n) = C'_1 + C'_2 + C'_3 + C'_4 + C'_5 \quad (11)$$

C'_1 is the total time cost of the operations before x;

$$C'_2 = \prod_{k=1}^{x-1} S_k T;$$

$$C'_3 = S_i T \sum_{m=x+1}^{y-1} \prod_{k=1}^{x-1} S_k (1 + \prod_{k=x+1}^{m} S_k);$$

$$C'_4 = \prod_{k=1}^{y-1} S_k T;$$

C'_5 is the total time cost of the operations after y.

It is obvious that $C_1=C'_1$, $C_2=C'_2$, $C_5=C'_5$ with the argumentation that $S_j<S_i$, we can infer that $C'_3<C_3$, $C'_4<C_4$. So we get the result that C' $(n)<C$ (n). It is not inconsistency with our assuming argumentation. So it is proofed.

4. CSOQP ALGORITHM ALANALYSIS

From Def. 2 and 3, we know that less selectivity and less time cost of processing each tuple, more quickly the operation is executed. However, we find that it is not fixed linear and nonlinear relation between the selectivity of the operation and the time cost of the processing each tuple. Just from Def. 2 and 3, it is not possible to address the problem about the selection of the query plans. In order to deal with the problem gracefully, we propose the CSOQP Algorithm.

We consider the two neighboring operations O_1 and O_2 in a query. Assuming that O_1 is executed before O_2, the number of the tuples entering into O_1 is a per second, their time costs of processing each tuple are T_1 and T_2 respectively, and selectivities of the two operations are S_1 and S_2 respectively. The total processing time cost of the two operations is $C=a(T_1+S_1T_2)$. If we change the positions of the two operations, the modified total time cost of the two operations is $C'=a(T_2+S_2T_1)$.

Because the operation O_1 and O_2 are neighboring operations, when we change their positions, other operations' cost don't change. Assuming that the modified query plan be better than the original query plan, in other words, $C'<C$. We can get the deduction below

$$\frac{1-S_2}{T_2} > \frac{1-S_1}{T_1} \quad (12)$$

So, We can conclude that to the any two neighboring operations, if they satisfy the condition Eq. (12), we will get the better query plan by reversing their positions. We get the definition 4 as follow:

***Definition* 4**. The initial CSOQP Algorithm is that the original query operation should be sorted descending by the result of the $(1-S_i)/T_i$.

PROOF: By using the reduction to absurdity, we proof the definition. So, if the conditions above are true, and the operations are not sorted descending by the result of the $(1-S_i)/T_i$, we can find at least one pair of O_i and O_j, which sequence number are x and y respectively and $x<y$ that O_i is executed before O_j and $(1-S_i)/T_i<(1-S_j)/T_j$. Because the two operations satisfy the condition Eq. (12), by reversing the two operations' positions, we can get the better query plan. It is not inconsistency with our assuming argumentation. So it is proofed.

Actually, within what we research above, we do not consider the relationship among the original operations. But in practice, we usually come into contact with the situation that they have some relativity. Taking an extreme example, there are two original O_1 and O_2, and they are the same operations. So the relativity between them is 1. Assuming that their selectivity is S, and the coming data rate is a per second, the left data rate should be aS, not aSS.

Considering the relationship among the original operations, we also consider the two neighboring operations O_1 and O_2 in a query. Assuming that O_1 is executed before O_2, the number of the tuples entering into O_1 is a per second, their time costs of processing each tuple are T_1 and T_2 respectively, the probability that the single tuple passing O_1 can pass O_2 is P_1, and the probability that the single tuple passing O_2 can pass O_1 is P_2. The total processing time cost of the two operations is $C=aT_1+aP_1T_2$. If we change the positions of the two operations, the modified total processing time cost of the two operations is $C'=aT_2+aP_2T_1$.

Because O_1 and O_2 are neighboring operations, when we change their positions, other operations' time cost doesn't change. Assuming that the modified query plan be better than the original query plan, in other word, $C'<C$. So we can get the deduction below:

$$\frac{1-P_2}{T_2} > \frac{1-P_1}{T_1} \quad (13)$$

So, we can conclude that to the any two neighboring operations, if they satisfy the condition Eq. (13), we will get the better query plan by reversing their positions. We get the definition 5 as follow:

***Definition* 5**. The common CSOQP Algorithm is that the original query operation should be sorted descending by the result of the $(1-P_i)/T_i$. P_i is the conditional selectivity which is the probability that the single tuple passing all operations

which order number is from 1 to $i-1$ can pass the i^{th} operation.

PROOF: We can proof it using the same method as using for Def. 4.

As to the fluctuant problem, we can introduce a fixed factor to ease the problem. If an operation exchanges its position with another operation, the operation should multiple δ, and by comparing the result of $\delta\ (1-P_i)/T_i > (1-P_j)/T_j$, we decide whether we reverse their positions or not. According to our experiment, it is proper to let the value of δ=0.8.

5. THE IMPLEMENT OF CSOQP ALGORITHM

The CSOQP Algorithm need to know the selectivity of the every original filtered operation or the conditional selectivity in the common CSOQP Algorithm and the time cost which is used every operation to process each tuple. To obtain the optimal query plan, theoretically, we must capture all tuple to reckon what we need. But it takes so much time and consumes so much memory that it is not tolerable. Therefore, we deal with the problem by using the sample method. Our architecture of the implement of the CSOQP Algorithm is showed as Fig.3.

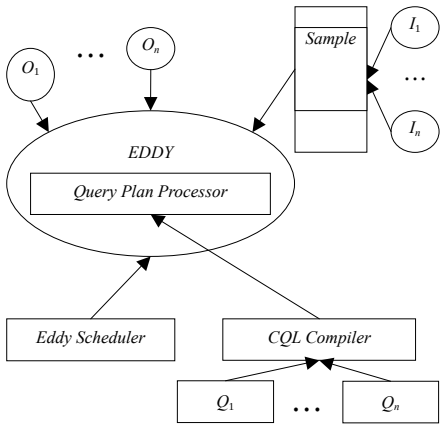


Fig. 3 The architecture of CSOQP Algorithm

Here we introduce the three modules, which are the CQL compiler module, the sample module, and the query plan processor module.

- *CQL Compiler Module*: We compile CQL query, get the original operations which the query contains, and get adjoining table *QILink* which reflects the relationship between the original operations *SUBQ* and the input stream ***I***.
- *Sample Module*: We capture the tuples in every the defining time interval and use *Tuple Warpper Sub-module* to encapsulate the tuples used to do the special process by sticking a *QP Tuple Shell* which indicates that those tuples must be routed to the *Query Plan Processor Module* to be dealt with.
- *Query Plan Processor Module*: When *Eddy* find the tuples, which have the *QP Tuple Shell*, these tuples are routed to the module. In the module, it records the starting and ending time, which is, used every operation to process each tuple, and we get the cost of every operation. Besides, the module keep the fixed number of the tuples status, to indicate the status that the tuples pass corresponding operation and the status that the tuples do not pass corresponding operation. For example, as to a query Q_i, the module keep the status of the sampling tuples by using the data structure called *TupleStatus*(Q_i) showed below. So in order to reckon the result of $(1-P_i)/T_i$ or $(1-S_i)/T_i$, we must know the result of the P_i or S_i. In our module, we use the other data matrix called P_I (Q_i) to indicate the number of the tuples which pass the original corresponding operations and the number of the tuples which pass preceding $i-1$ operations and also pass the i^{th} operation for the i^{th} row, showed as Fig.4. From Fig.4, we can easily reckon the result of the P_i and S_i respectively and we get the nearly optimal query plan.

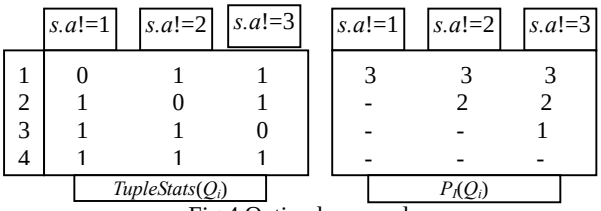


Fig.4 Optimal query plan

6. CONCLUSION AND FUTURE WORK

We have show that CSOQP Algorithm has the strong guarantees in theoretically and it also has good performance in experiment. CSOQP Algorithm handles correlated filtered operation in environments where changes may be rapid and unpredictable.

An interesting avenue for future work is the problem of balancing query execution against profiling and reoptimization overhead for optimal performance in an online adaptive setting.

REFERENCES

[1] R. Avnur, J. Hellerstein, "Eddies: Continuously adaptive query processing", Proc. of the 2000 ACM SIGMOD Intl.Conf. on Management of Data, May 2000, 261-272.

[2] A. Deshpande, "An initial study of overheads of eddies", SIGMOD Record, Dec. 2003, 32(4).

[3] T. Urhan, M. Franklin, "Xjoin: A reactively-scheduled pipelined join operator", IEEE Data Engineering Bulletin, May 2000, 23(2): 27-33.

[4] P. J. Haas, J. M. Hellerstein, "Ripple Joins for Online Aggregation", SIGMOD Conference, 1999, 287-298.

[5] J. Kang, J. F. Naughton, et al, "Evaluating window joins over unbounded streams", ICDE 2003, 37-48.

[6] N. Tatbul, U. Cetintemel, et al, "Load Shedding in a Data Stream Manager", VLDB, 2003.

[7] B. Babcock, M. Datar, et al, "Load Shedding Techniques for Data Stream Systems", MPDS Workshop 2003.

A New Collaborative Filtering Algorithm for Recommender Systems

Yao Yu, Shanfeng Zhu*, Jinshuo Liu, Xinmeng Chen
Computer School, Wuhan University, Wuhan Hubei 430072, China
Email: yaoyu928@163.com
***Institute for Chemical Research, Kyoto University, Uji Kyoto 611-0011, Japan**

ABSTRACT

Kendall correlation based collaborative filtering algorithm was proposed for the recommendation problem. The Kendall correlation method is used to measure the correlation amongst users, which considers only the relative order of the ratings. Although the experiments show that Kendall based algorithm performs slightly worse than the classic Pearson algorithm, it is based on a more general model and thus could be more widely applied in e-commerce. Another discovery is that the consideration of only positive correlated neighbors in prediction, in both Pearson and Kendall algorithms, achieves higher accuracy than the consideration of all neighbors, with only a small loss of coverage.

Keywords: Kendall correlation, collaborative filtering, recommender system, positive correlation, data mining.

1. INTRODUCTION

With the development of E-commerce, recommender systems are widely employed by Web sites to assist consumers in finding desired products by suggesting products, which are predicted to be interesting to the consumers. Typical recommendation approaches include expert advices, statistical information, data mining approaches, content based methods, and collaborative filtering. In this work, we will study collaborative filtering approaches that recommend products to active user based on the opinions of other users. We focus on memory-based collaborative filtering approaches. The most popular memory-based collaborative filtering algorithm is a kind of neighborhood based algorithm consisting of three steps:

• Correlation calculation: The correlations (or similarities) between the active user with any other users are calculated.
• Neighborhood selection: after calculating the correlations of the users to the active user, choose the neighborhood of the active user in two ways: threshold or top-N.
• Prediction: once the neighborhood has been determined, it is used to predict the active user's rating on new items.

1.1 Related Work

The differences in neighborhood based collaborative filtering algorithms lie in the methods adopted in these three steps, especially in the ways of calculating correlations. Main techniques for correlation calculation include vector similarity, mean square difference, Pearson correlation, constrained Pearson correlation, Spearman rank correlation and entropy-based correlation. Pearson correlation method, which comes from statistics, calculates correlation based upon the average deviation from mean rating, and was first utilized in GroupLens project [1]. Constrained Pearson correlation method, which was introduced in Ringo music recommendation system [2], is a variation of Pearson correlation method. In contrast to Pearson correlation method, Spearman rank correlation calculates the correlation by considering the ranks of the ratings instead of the values of the ratings [3]. Mean square difference, which was also introduced in the Ringo system [2], assigned the correlation according to mean square difference between each pair of users. Several studies on empirical analysis of collaborative filtering algorithms reported that classical Pearson correlation algorithms outperforms vector similarity approach, mean square difference approach and Spearman rank correlation approach in the experiments [4].

1.2 Motivations and Contributions

All these algorithms for correlation calculation are rooted in different mathematics models and thus subject to corresponding assumptions [3]. Since users rate items in a very easy way without rigorous standards, these ratings would definitely violate these assumptions. As we know, the user's rating usually has many grades (such as 5 or 7 scale), and user prefers high rated items to low rated items, it should be more suitable to view these ratings as ordinal variables than scale variables. If users rate items in different ranges, it will be more obvious. To compute the correlations among ordinal variables, Kendall correlation is frequently discussed in statistics literature [5,6].

We make use of Kendall correlation algorithm to form recommendation. The performance of the algorithm is compared with Pearson correlation algorithm by conducting experiments on real data. Kendall correlation algorithm has less restriction on underlying data than Pearson algorithm, thus has wider application, and that performs only slightly worse than Pearson correlation algorithm. Different with the similarity and distance, correlation between user pair can be either positive or negative. To formulate the neighborhood of active user, other users that are highly correlated with active user are selected for future prediction. Some systems, such as GroupLens, chose both negative and positive correlated neighbors to form neighborhood, while some other systems only selected highly positive correlated neighbors. In experiments we compare the roles of positive and negative correlated neighbors in generating prediction.

2. NEW CORRELATION ALGORITHM

2.1 Kendall vs. Spearman

Although both Spearman and Kendall are proposed to deal with correlations among ordinal variables, they differ in computing approaches, underlying assumptions and thus scope of applications. Spearman rank correlation uses the numerical difference of ranks for each rating. Furthermore, Kendall uses only the relative order of ranks, thus is more nonparametric than Spearman, and has an intuitively simple interpretation than Spearman. The absolute value of Kendall coefficients tends to be slightly smaller than Spearman's coefficient for the same set of data. If there are only a few possible values for rating, many items will be assigned in the same grade. That is, there are ties among the ranks. This

is exactly the situation in recommender system where thousands of items are rated in a relatively small range. Under these circumstances, Spearman is not very natural, since it will assign the items with same rating a mid-rank value that is not very meaningful in reality, whereas Kendall is still quite natural. And it was reported that Kendall deals better with ties than Spearman [6]. Considering these characteristics, we will make use of Kendall method for correlation calculation in this work.

Here we introduce how to calculate Kendall correlation coefficients and its extension for tied ranks, which is often discussed in statistics literature [5]. Firstly, we discuss the case where there are no ties among ranks [6]. Given a set of items $N = \{1, 2, \ldots, n\}$, a ranking τ with respect to N is a permutation of all elements of N which represents a user's preference on these items. For each $i \in N$, $\tau(i)$ denotes the position of the element i in τ, and for any two elements $i, j \in N$, $\tau(i)<\tau(j)$ implies that i is ranked higher than j by τ. The Kendall-τ distance between two rankings τ_1, τ_2 is defined as: $D(\tau_1, \tau_2) = |\{(i,j): \tau_1(i) < \tau_1(j) \wedge \tau_2(j) < \tau_2(i), \forall i,j \in \boldsymbol{N}\}|$. Usually it will be normalized by dividing the maximum possible distance $n * (n - 1)/2$. Similarly, denote $C(\tau_1, \tau_2)$ as the concordance between these two rankings. That is,

$C(\tau_1, \tau_2) = |\{(i,j): \tau_1(i) < \tau_1(j) \wedge \tau_2(i) < \tau_2(j), \forall i,j \in \boldsymbol{N}\}|$.

In contrast to Kendall-τ distance, Kendall-τ correlation coefficient is calculated by counting the difference between the numbers of concordant pairs with the number of discordant pairs in these two rankings. That is,

$$K_a(\tau_1, \tau_2) = \frac{C(\tau_1, \tau_2) - D(\tau_1, \tau_2)}{n(n-1)/2} \qquad \textbf{Eq.(1)}$$

In case of full ranking without tied ranks, we can easily get $C(\tau_1, \tau_2) = n(n - 1)/2 - D(\tau_1, \tau_2)$. Thus,

$$K_a(\tau_1, \tau_2) = \frac{n(n-1)/2 - 2D(\tau_1, \tau_2)}{n(n-1)/2} = 1 - \frac{4D(\tau_1, \tau_2)}{n(n-1)} \qquad \textbf{Eq.(2)}$$

Secondly, if there are ties among ranks, we must extend original Kendall correlation definition to these ties [6]. These situations are unavoidable while putting a large number of items into several levels in case of recommendation problem. The extension is proposed by considering ties in ranks when comparing relative order of item pairs in different rankings. We denote T_{τ_1} as the number of ties on the τ_1 while comparing, and T_{τ_2} as the number of ties on the τ_2. Then the extended Kendall correlation coefficient is defined as: $K_b(\tau_1, \tau_2) =$

$$\frac{C(\tau_1, \tau_2) - D(\tau_1, \tau_2)}{\sqrt{C(\tau_1, \tau_2) + D(\tau_1, \tau_2) + T_{\tau_1}} * \sqrt{C(\tau_1, \tau_2) + D(\tau_1, \tau_2) + T_{\tau_2}}} \qquad \textbf{Eq.(3)}$$

The value of K_b lies between -1 and 1 that stands for complete rank agreement or disagreement respectively.

2.2 Kendall Correlation Method

1) Correlation calculation

We use extended Kendall correlation discussed above to tackle heavily tied ranks in the case of recommendation problem. We choose this method because we think that the characteristic of rating given by user is ordinal more than scale, and Kendall correlation is just proposed for measuring the correlation between ordinal variables.

2) Neighborhood selection

We adopt correlation weight threshold method, and find that the coverage doesn't decrease much if we consider all possible neighbors. And we impose another restriction, co-rated threshold, on the users acting as neighborhood of the active user. That is, the number of co-rated items must be greater than a threshold. The reason for this restriction is that the higher the number of co-rated items between selected user with the active user, the more reliable the correlation is. The specific value of the threshold is determined by practical problem, which will be discussed in later section.

3) Prediction generation

The prediction method inherits classical Pearson methods that calculate average deviation of a neighbor's rating from the neighbor's mean rating over all the items rated by the neighbor. That is,

$$p_{a,j} = \overline{v_a} + \frac{\sum_{i=1}^{s} K_{a,i}(v_{i,j} - \overline{v_i})}{\sum_{i=1}^{s} |K_{a,i}|} \qquad \textbf{Eq.(4)}$$

Of course, the correlation coefficient $K_{a,i}$ between user i and active user a is computed by Kendall correlation.

3. EXPERIMENTS AND RESULTS

3.1 Experimental Procedure

To explore the performance of our algorithm, we need to carry out experiments on real data. In this work, the data is provided by the Grouplens project [1]. This data set is collected in the MovieLens (http://movielens.umn.edu) web site for movies recommendation in a seven-month period. After some preprocessing, e.g. the removal of users rating less than 20 movies, the data set consists of 100,000 ratings from 943 users on 1682 movies. The ratings given by the users range from 1 to 5, where 5 stands for the best level. For predicting active user's rating on some selected items, we adopt the same strategy as the one in [4], that is, all users are selected as active users respectively, and for each chosen user, withheld ratings on 10 items for prediction. After the active user's ratings on target items are predicted, the performance of algorithm is measured by comparing the predicted ratings and real ratings given by users.

Here we make use of two key metrics, coverage and accuracy, for performance evaluation. Coverage measures the percentage of ratings on items, which could be predicted by the algorithms, and accuracy metric measures the correctness of prediction. We make use of MAE as the accuracy evaluation metric. Given n target rating on items for prediction, if q_i is the rating given by the users on item i, and p_i is the rating predicted by algorithms, the mean absolute error of the algorithms is formulated as:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \qquad \textbf{Eq. (5)}$$

The lower the MAE, the more accurately the algorithm predicts the ratings on the target items.

At first, the correlations between active user and any other users are computed respectively by Pearson and Kendall method. Then for the neighborhood selection, we set up a unified condition. That is, the users in the neighborhood of active user should have co-rated at least 6 items with active user, and the absolute correlation between the active user and users in neighborhood should be greater than a threshold value. Adopt 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45 and 0.5 as threshold respectively in a series of experiments. The threshold value 6 is determined by the characteristic of the data. In experiment, after withholding 10 ratings for each user, select any two users and measure the number of the co-rated items between these two users. In each round, analyze $C_{943}^2 = 444153$ pairs of users and compute the mean, median, mode, and minimum, maximum of the number of co-rated items. After analyze 100 rounds of data, we get that the median number of co-rated items of any two

users in these rounds are 6. In other words, about 50% of user pairs co-rate more than 6 items. Thus we suspect the reliability of correlation coefficients calculated when the number of co-rated items is less than 6. Of course, we can set a high value, but it will decrease the coverage of the predictions.

3.2 Results of Pearson vs. Kendall

In the first experiment, we compare the performance of these two algorithms by coverage and MAE. In each round, we predict the same target ratings by both algorithms with different correlation thresholds, and then performance metrics are computed and averaged for the whole rating set. We run altogether 100 rounds, and need to predict 943,000 ratings. Finally the mean of these two measures for all rounds are recorded for comparison. Fig. 1. shows the comparison of MAE and coverage of these two algorithms based on the experiment result data. We could see that they behave similarly in various aspects. With the increase of correlation threshold, the coverage of both algorithms will decrease. For accuracy, with the increase of the weight threshold, the MAE will first decrease, then increase after reaching a lowest point. It means that high correlated neighbors don't predict target rating effectively. This result is consistent with other researcher's result, such as [4]. Additionally, the significance of difference of MAE between Pearson and Kendall is 0 at each correlation threshold, which is calculated by statistical software SPSS10.

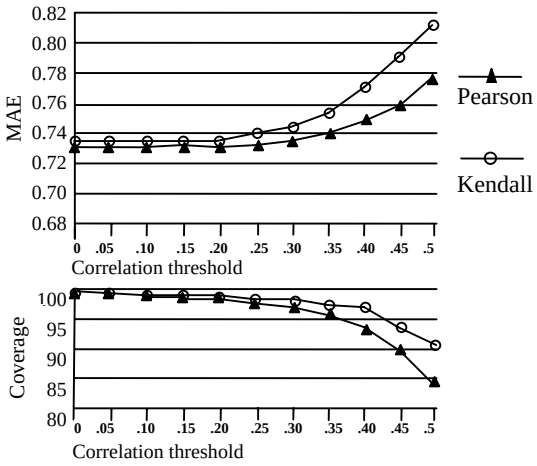


Fig. 1. Performance comparision on MAE and coverage of pearson and kendall

3.3 Positive vs. Negative Correlated Neighbors

We want to compare the effectiveness of positive correlated and negative correlated neighbors in collaborative filtering prediction. In the first experiment, the users in neighborhood of active user for prediction contain both positive correlated users and negative correlated users. In order to discover their distinct roles in prediction, we design three groups of users for prediction in this experiment: only positive correlated users, all users, and only negative correlated users. We also carry out 100 rounds of test, but the correlation weight threshold is assigned 0, 0.05, 0.1, 0.2, 0.3, 0.4, and 0.5 respectively. Fig. 2. shows the parts of experimental result. Here we denote C_P, C_N and C_A as the coverage of positive, negative correlated and all neighbors prediction respectively. Similarly denote M_P, M_N and M_A as the mean absolute error of positive, negative correlated and all neighbors prediction. In all cases of different thresholds, we find that $C_A > C_P > C_N$, and at the same time $M_P < M_A < M_N$. It means that incorporating negative correlated neighbors in prediction increases the coverage of algorithms, but also decreases the accuracy of the results. From the result data, can find that the difference among negative correlated, positive correlated and all correlated neighbors for prediction is significant from the point of view of statistics. Due to limited space, we don't present related table here.

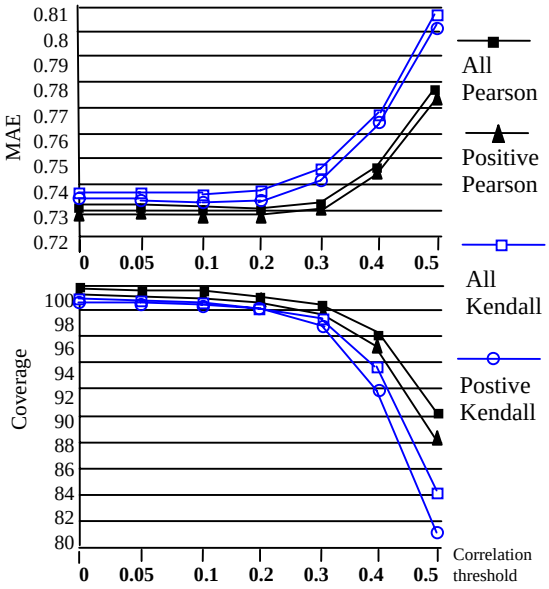


Fig. 2. Performance comparision on positive vs. all related neighbors

4. CONCLUSIONS

In this paper, we propose Kendall correlation based collaborative filtering algorithms for recommender systems in e-commerce. Kendall correlation algorithm has more reasonable theoretic background than Pearson based algorithms, thus could be applied more naturally in e-commerce. Furthermore, for neighborhood based collaborative filtering algorithms, we find that only considering positive correlated neighbors in prediction achieves higher accuracy than considering all neighbors, with only a small decrease in coverage.

REFERENCES

[1] P. Resnick, N. Iacovou, et al, "Grouplens: An open architecture for collaborative filtering of netnews", Proceedings of ACM 1994 Conference on CSCW, 1994, 175-186.

[2] U. Shardanand, P. Maes, "Social information filtering: algorithms for automating 'word of mouth' ", Proceedings of ACM CHI' 95 Conference on Human Factors in Computing Systems, 1995, 210-217.

[3] J. Herlocker, J. Konstan, et al, "An algorithmic framework for performing collaborative filtering", SIGIR, 1999, 230-237.

[4] J. Herlocker, J. A. Konstan, et al, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms", Information Retrieval, March 2002, 287-310.

[5] M. Kendall, J. D. Gibbons, Rank Correlation Methods. Oxford University Press, fifth edition, 1990.

[6] N. Cliff, Ordinal Methods for Behavioral Data Analysis. Lawrence Erlbaum Associates, Mahwah, New Jersey, 1996.

A Sliding Window Algorithm for Mining Frequent Itemsets on Data Stream

Junqiang Liu
Institute of Artificial Intelligence, Zhejiang Gongshang University
Hangzhou, Zhejiang, China
Email: ljq18@hotmail.com
And
Xiurong Li
College of Computer and Information Engineering, Zhejiang Gongshang University
Hangzhou, Zhejiang, China
Email: lixiurong7982@hotmail.com

ABSTRACT

Some applications generate continuous, unbounded, and extremely fast data stream. Due to its characteristics, it is difficult to apply the existing productions of mining frequent itemsets in traditional databases to the data stream environment directly. A one-scan algorithm SFP of mining frequent itemsets on data stream was proposed by using a fixed-size sliding window model and adopting a data structure of SPFT. The window limited memory usage. SPFT made use of the advantages of FP-Growth and additionally introduced a pivotal timeIdList into the SPFT. TimeIdList marked the itemsets contained in those expired transactions so as to remove the effect of old transactions on the mining results as the window slides forward.

Keywords: Data Mining, Data Stream, Frequent Itemsets, Sliding Window.

1. INTRODUCTION

Recently a new data type—data stream is found in many application domains including network traffic analysis, web click stream mining, sensor network data analysis, dynamic tracing of stock fluctuation, and telecom records analysis. A data stream is an ordered sequence of items that arrives in timely order. Different from data in traditional static databases, data streams are continuous, unbounded, usually come with high speed and have a data distribution that often changes with time.

Data streams differ from the conventional stored relation model in several ways [6]:

The data elements in the stream arrive online.

The system has no control over the order in which data elements arrive to be processed, either within a data stream or across data streams.

Data streams are potentially unbounded in size.

Once an element from a data stream has been processed it is discarded or archived, it cannot be retrieved easily unless it is explicitly stored in memory, which typically is small relative to the size of the data stream.

Due to the characteristics of data stream and the constraints of limited system source, there is neither enough space to store all the data nor time to rescan data again. So it is always impossible to produce exact answers.

Mining frequent itemsets is the essential problem in data mining. Although there have been many productions of frequent itemsets mining in traditional database, it is very challenging to extend them to data stream environment. The first algorithm is Apriori [2]. After that, many other algorithms based on Apriori are developed for performance improvement [3]. But Apriori-based algorithms require multiple scans of the database therefore they are not suitable for a data stream environment in which data can be scanned only once. Another classical algorithm FP-Growth proposed in [4] using a frequent pattern tree (FP-tree) data structure and an FP-Growth algorithm without generating candidate itemsets. However it still needs two scans of data, so it can't be directly used to mine frequent itemsets in data streams either.

In many cases, we are more interested in the frequent itemsets mined recently. Sliding window is a technique which emphasizes recent data rather than the entire past history of the data stream for producing an approximate answer. The size of the sliding window specified as the number of records may be decided according to applications and system resources. The mining result of the sliding window method totally depends on recently generated transactions in the range of the window.

The fundamental problem is that as the window slides forward, old results must be simultaneously evicted and new ones generated from new transactions are appended. So we need to store some information about the time signs of the transactions in the window in order to discard those itmesets, which are out of the range of the window.

In this paper, inspired by the fact that FP-tree provides an effective data structure for frequent itmesets mining, we develop a one-pass algorithm Stream-FP(abbreviated **SFP**) based on a sliding window model of data stream. Here we take a FP-tree-based structure, the radical difference of which is that we introduce a field *timeIdList* (described in Section 3) to distinguish the order of transaction in order to release the effect of stale data on mining results.

The rest of the paper is organized as follows: Preliminaries are given in Section 2, algorithm SFP is described in detail in Section 3, and Section 4 concludes the paper.

2. PRELIMINARIES

For finding frequent itemsets, a data stream can be defined as follows:

i) Let $\boldsymbol{I}=\{i1, i2,..., im\}$ be a set of current items

ii) A transaction is a subset of ***each transaction and I*** have a unique transaction identifier *TID*.

iii) Let N be the total number of transactions in data streams, which have arrived until now.

iv) Let w be the size of sliding window, which means that only recent w transactions are in the range of the current window.

v) Occurrence frequency of an itemset is the number of transactions that contain the itemset, which is also known as support counter or frequency. If the frequency of an itemset

is greater than or equal to the specified minimal support min_{sup}, then it is a frequent itemset.

vi) If some itemsets, which at some period may be infrequent, become frequent in the near future, we call them significant itemsets. In order to improve the accuracy of approximate mining results, we should store their information in the data structure too. Only those insignificant itemsets whose counters are much less than min_{sup} are pruned to save space usage. So we introduce a $min_{pru,\ which}$ is less than $min_{sup,\ and}$ use min_{pru} instead of min_{sup} during some pruning operations (described in Section 3).

Fig. 1. gives us a fixed-size (w=5) sliding window model of a data stream.

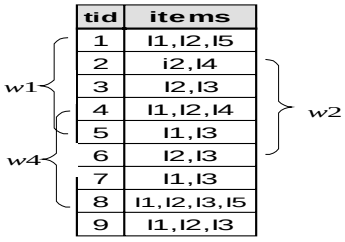


Fig. 1. The sliding window model of the data stream, w=5

3. ALGORITHM SFP

3.1 Data structure

In our algorithm we use a data structure called **SFP**, an efficient FP-tree-based structure for maintaining the crucial information in the memory to mine frequent itemsets from data streams. It consists of two components. One is a Stream-FP-tree (**SFPT** in short) and the other is a header table (**SHT** in short). They are described in detail below.

(1) SFPT stores the compressed information of the data streams in the current window. Each node in the tree includes four fields: *itemName, counter, nextLink,* and *timeIdList*. Here, *itemName* registers which item this node represents, *counter* registers the number of transactions represented by the portion of the path reaching this node, *nextLink* links to the next node in the **SFPT** carrying the same *itemName* or null if there is none, and *timeIdList* indicates the list of *timeId* when it enters this node. *timeId* is calculate as follows, the total number *N* of transactions mode *w*, for example, if *w*=3, and the item *i* enters a certain node when the fourth transaction arrivals, then timeId =1 (4 mod 3) and 1 should be appended in the field of *timeIdList*.

(2) SHT is similar to FP-tree's header table. Each entry (*itemName, counter, link*) in the table denotes the name of one item, the total number of the item in the current window and the pointer which points to the first node carrying the same *itemName* in the **SFPT** respectively.

The algorithm includes two phases. The first is to construct and dynamically update a **SFPT,** and the second is to mine the frequent itemsets from the tree.

3.2 Algorithm

The algorithm of constructing and dynamically updating the SPFT is outlined below.

1) Constructing a SPFT

This step is called initialization process. It happens when *N* is less then or equal to *w*. When a new transaction is generated, we only need to insert those itemsets induced from this transaction into the tree without considering the effect of the previous arrived transactions because they are all in the range of the current window.

1). Initializing the tree: Create the root of the tree by setting it as "null".

2). When a new transaction with items sorted in lexicographic order is generated, insert each item *e* to the tree in the same way proposed in [4], with the only difference that the node has an additional field labeled *timeIdList.* And construct or update the SHT simultaneously by adding a new entry (*e, 1, link*) when there is no entry whose itemName is *e* otherwise updating the corresponding entry by increasing its counter by one.

Fig. 2. shows the SPT and the SPT in the range of w1.

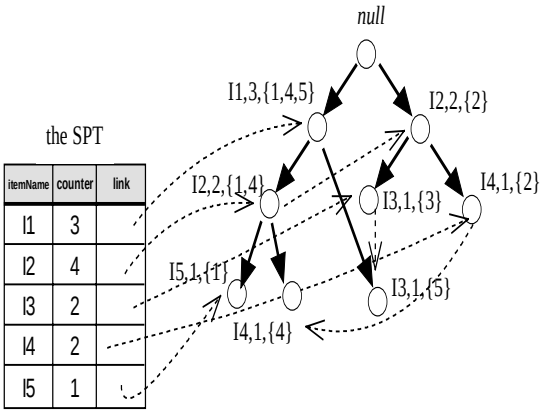


Fig. 2. The SPFT of w1

2) Updating the SPFT

Subsequently we enter the step of the window sliding. While a new transaction is arriving, the earliest one is obsolete. We need to update the in-memory data structure to remove the obsolete transaction and add the new one; we name them removing and adding operation respectively. They are expressed as follows.

(1) Removing operation

First, calculate the *timeId* of obsolete transaction recorded in the *timeIdList*, *timeId* = *N* mod *w*, then searches those nodes whose *timeIdList* field includes *timeId*, and does as follows:

i) Set node. *counter* = *counter* – 1, remove *timeId* from *timeIdList*.

ii) Updating SHT: look for the corresponding entry and set *counter* = *counter* – 1, *link* = node.*nextLink*. Sspecifically, if the updated counter decreases to 0, then delete the entry from table to save space.

(2) Adding operation

Read the newly generated transaction, insert its items into the tree in the way of constructing a **SPFT** above.

In the Fig. 3., we show the updated **SPFT** of w2, here *N*=6, *timeId* =1. Firstly, during the removing operation, the *timeIdLists* of nodes {*I1, I2, I5*} contain 1, so the *counters* of these nodes are decreased by one and 1 is removed form *timeIdList*, and in the corresponding entry of **SPT** all the counters are decreased by one too. Specifically, the updated *counter* of I5 in the SPT becomes 0; consequently it was deleted from the table. Secondly, it is time to perform the adding operation for the last transaction 6, because I2 has existed in the tree, its counters of the entry and the node are increased by one, and 1 is appended to *itemLIst*.

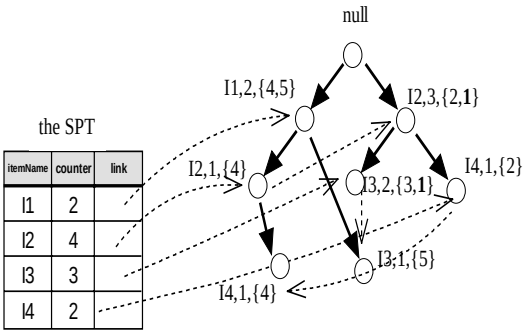


Fig. 3. The SPFT of w2 updated from Fig. 1

3) Mining Frequent Itemsets From The SPFT

During the processes above, we may prune those infrequent items whose counter is less then min_{pru} whenever we need to save space and improve the efficiency. The pruning operation is depicted as follow.

Pruning operation

Look up the entries in SHT whose counters are less than min_{pru} , and then delete them from the table as well as those nodes in the tree linked by the filed *link*.

Finally, when we need to select all the frequent itemsets so far, we first perform the pruning operation and then begin mining the tree.

Here we mine the tree in the same way proposed in [5] which explores the tree in the top-down order and constructs not conditional pattern bases and sub-trees as in [4] but sub-tables to save substantial amount of time and space.

Fig. 4. gives the pruned SFPT from Fig. 3.

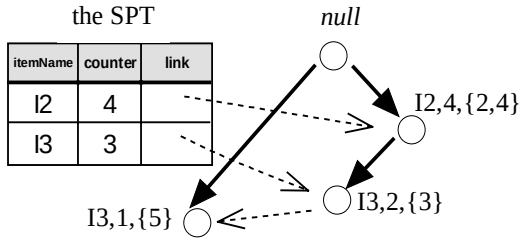


Fig. 4. The pruned SPFT of w2

4. CONCLUSIONS

In this paper, we propose an algorithm based on the fixed-size sliding window model and adopt the data structure similar to FP-tree structure. We make the best of the production of FP-Growth algorithm about mining frequent itemsets in traditional static database and extend it to the data stream mining.

We still have some aspects in the algorithm for improvement, such as considering changeable-size sliding window model and time-based sliding window containing transactions which have arrived in the last T time unit, considering more processing speed problems, etc.

5. REFERENCES

[1] J.W.Han, Miceline Kamber, Data Mining: Consepts and Techniques, China Machine Press, 2001.

[2] R. Agrawal, T. Imielinski, et al, “Mining Association Rules between Sets of Items in Massive Databases”, Proceedings of the ACM SIGMOD Conference on Management of data, 1993, 207-216.

[3] R. Agrawal, R. Srikant, “Fast algorithms for mining association rules”, Proceedings of the Very Large Data Bases (VLDB’94), Chile: Santiago de Chile, 1994, 487-499.

[4] J.W.Han, J. Pei, et al, 2000, “Mining frequent patterns without candidate generation”, Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), 1-12.

[5] K.Wang, T.Liu, et al, “Top Down FP-Growth for Association Rule Mining”, Proceedings of the Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2002), Taipei, 2002, 334-340.

[6] B. Babcock, S. Babu, et al, “Models and issues in data stream systems”, Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Wisconsin: Madison, 2002, 1-16.

[7] H.F.Li, S.Y. Lee, et al, “An Efficient Algorithm for Mining Frequent Itemsets over the Entire History of Data Streams”, Int'l Workshop on Knowledge Discovery in Data Streams, 2004, 20-24.

[8] J.H.Chang, W.S.Lee, “A Sliding Window Method for Finding Recently Frequent Itemsets over Online Data Streams”, Journal of Information Science and Engineering, 2004, 20(4), 753-762.

The Performance Test and Evaluation of Distributed Data Mining

Min Zhang
Computer Science and Information Engineering, Zhejiang Gongshang University
Hangzhou, Zhejiang, China
Email: zhangmin629@163.com
Hongxia Shi
Computer Science and Information Engineering, Zhejiang Gongshang University
Hangzhou, Zhejiang, China
Email: shihx@mail.zjgsu.edu.cn

ABSTRACT

Currently the direction of data mining tends distributed and parallel calculation. In order to evaluate their performance, a test system structure that can test and evaluate performance of data mining system is present in this paper. This system is based on excitation response mechanism: the control center of test defines test events, designs test packages, and sends the test information to every tested subsystems; agents are motivated to analyses instruction, distill and return data to evaluation system. Data collection subsystem incorporates the result data with its particular method. A particular after-event evaluation mode is present to estimate whether data mining is good or not. Within performance evaluation indexes, such as data relative precision, result cover degree, and collection velocity can gets the last evaluation result.

Keywords: distributed, data mining, performance test, data collection, methods of evaluation.

1. INTRODUCTION

Distributed data mining and application research has been a major task during more and more type of data can be obtained. Specialists and scholars propose numerous patterns and algorithms of the distributed data mining. Distributed data mining means fleetly and effectively collect connotative and useful information or models from different places or sites having same or different structures of databases, files or pages. Results of data mining can supply decision support. In mining process, system collects useful unknown information within one or more character. So every Web site must transform its partly data sets into a fitting form as a whole regulation, then dispenses them to every tested subsystem by the network; Multi-Web site performs mining algorithm simultaneously to decrease the load of background server. System distills information that be carried on centralized data. At last, system completes knowledge study and consequence process.

Now various models of distributed data mining have therefore been developed, many of which are related to isomorphic or isomeric algorithms according to data category and storage location. The isomorphic distributed mining, which data are provided with a uniform logical pattern but geographical positions are on the distributed different sites; the isomeric is that its features or attributes are always not same. But whether the isomorphic or isomeric, the data can shift dynamically in space and time. Process of isomeric data mining is: the server module accepts the request of client-side program; In order to really implement the distributed function, that server module ought to be required to broadcast the surrounding node; Mining modules return the result data with some protocol (such as HTTP, SOAP) according with the demand. The server module receives the data, handle them suitably, and integrate to a database.

2. AN ARCHITECTURE OF MINING TEST

The use of test technique, in particular, has become an essential part of data mining studies. The results of data mining have direct effect on the data validity of decision support system. Therefore, how we can judge whether distributed data mining is good or not? On the basis of this, the author proposes an ordinary architecture of the distributed data mining. In this architecture, the test system is separated into three modules: first module is the testing control manager, it defines test event and designs test point, also supervises the performance of test; the second is dynamic subsystem that being distributed in all the places, which mines required data of test system, at the same time, the module also calculates needed time. Third module, using analysis and evaluation system, are therefore, likely to analyze the data and information that are collected by tested system and put them into analysis center and evaluation manager. Its composition in this system is shown Fig. 1..

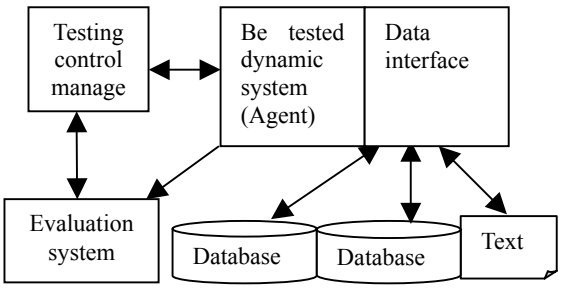


Fig. 1. The architecture of system

In the architecture outlined above, whole system is divided into part of test and part of evaluation. The part of test concludes test control manager and Agent (be tested dynamic subsystem), and the other (which is evaluation) consists of the data collection subsystem and the evaluation subsystem. The data collection subsystem chiefly is in charge of the data gathering, data collation, data filtration and certification. Yet the evaluation subsystem takes charge of forming evaluating index and establishing evaluating pattern to get results of evaluation.

3. TEST SYSTEM

3.1 Testing control manager

The testing control manager is the testing control center, defines instruction sequence and parameter that capturing performance data that the definition system program that will carry on the performance analysis, The relevant test instructions are sent to Agent, and the acceptance tests return data, then these data are sent into the evaluation system. It chiefly consists of:

(1) Test instruction customization: defining an ordinary form of the parameter testing instruction, because the act of system is abstracted, we can look upon the instruction as a series of events gatherings, and analyze the data above the result. The main parameter in the test instruction is: the initial and end port address, concrete records, the repeated records, instruction sent time and received time, size of cache, test instruction size, test range.

(2) Test instruction administration: preprocessing to the system, selecting the parameter scheme, sending the test instruction to every Agent in the range of test, returning the signal handle after dealing with every Agent's calculation, carrying on evaluating performance analysis by the evaluation system.

3.2 Tested dynamic subsystem (Agent)

While distributed Agent receives the instruction and parameter, system put them into analysis and transform, whose transform type is required or needed. It must perform test instructions and collect data from different ports, every Agent calculates the time between sending the request and returning the data. The records, making of data, are organized with specified requirement and returned to the evaluation system finally.

We adopt an event driven method in the test. The test event, being beforehand defined and certificated, is allocated an event identifier, that is, event code, which represents an event sort. The event ordinarily may include three types: simple event, complex event and intelligent event, they are called logic event. A logic event is made of one event code and some fixed-length or unfixed-length event substances. As to simple event, just easy query and collection, we should confirm event code and the simple event substance; as to complex event, divide or transform event substance into several simple event data after complex event code, and to intelligent event, there is an unfixed-length character string immediately subsequent to the event code. Applying various methods of deducing methods such as the semantic, regulation, the classification and relation and so on to extend the event substance. Every large or small test instruction is assigned by the call program to data mining subsystem, then runs, tests and analyzes the program, which target is that enlarges or confines the object to some limit, afterwards, in this limit, running and analyzing, causing smaller circle around to analyze performance, mend system and improve performance.

In tested system (Agent), when receiving module receives the test instruction, it notifies analysis module to confirm the type of mining on the basis of event code, at the same time, transforms test request into needed form, ties the settled interface against the server about the test database and other file. In transform and analysis time，getting the knowledge of correlative movement information to be analyzed and deduced. After local mining and returning data, maybe at the same time, Agent's knowledge base is rich on the basis of the relevant form of mining movement. The concrete systematic flow diagram is shown Fig. 2..

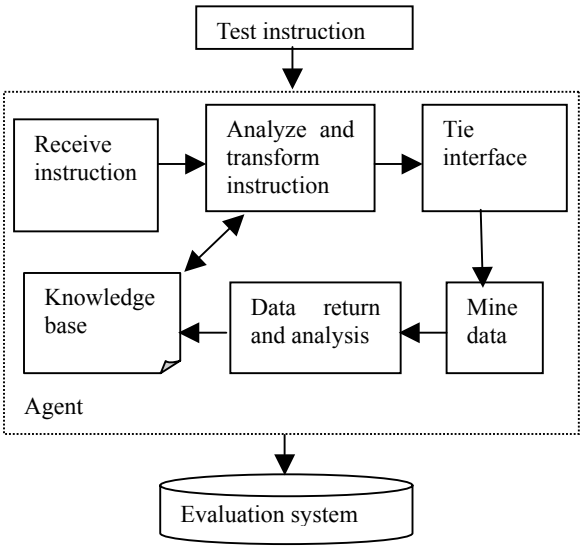


Fig. 2. Systematic flow of tested subsystem

4. EVALUATION SYSTEM

The collection subsystem is utilized by the evaluation system that collects with the data, after which, the system goes into evaluation subsystem. Evaluation subsystem gets the last results to estimate whether this system is good or not depending on the super or inferior data, these data are all analyzed by evaluation model. The evaluation system can be divided into the data collection subsystem and the evaluation subsystem.

4.1 Data collection subsystem

Incorporation of data: Much subroutine will come into being much of test event data, every event data first of all will be executed incorporation function according to the test event time and the parameter identifier, taking shape an integrated logic event data in a period of time. This is the base of performance analysis and evaluation.

Data filter and separation: The integrated logic event data are all from correlation events in the tested system, by comparison, the total quantities of data are very great, the interior event relationship is complex, so it is necessary that filter and separate the integrated event flow on the basis of every relative event point and the abstract level. For instance: in the intelligent test, the data is simplified out from some correlative test vertically, obtaining process level average performance time and the relationship between event and event.

Data certification: Before being to evaluate, it is necessary that check whether the data is valid or not, whether the event record sign is valid or not and the event data is well ordered or not. This is indispensable step for ensuring the legitimacy of performance analysis and evaluation.

4.2 Evaluation subsystem

Indexes of the evaluation construction: After completing data incorporation, the evaluation of performance is carried out. The data in the evaluation are from analysis database. The evaluation of performance model may relate to a number of aspects, but this paper only involves a simplified model, which is chiefly composed of some key field of

performance parameter and data collecting parameter, and which chooses the difference parameter to reflect different performance, then draw up different evaluating level. It mainly considers several factors as follows:
(1) Data accuracy of result, when query information, not fully concerns to how many of results to be returned, but see whether the results are coincide with the demand of requirement. It still is concerned how correlative degree between collected data and required data arrives to.
(2) Data extent, it just means data cover degree, it indicates that data resources ought to be covered as far as possible similar to the given associated theme word.
(3) Collection velocity, which is the number of collected data in the time as far as possible brief. (Network response time, the package lost rate, delay time, network throughput).

4.3 Methods of evaluation

Measure method of relative accuracy Quality: Measure as to one event or simple event may be evaluated by the method of mining relative accuracy. The relative accuracy means degree of mined data validity and correlation, and they are separated into three ranks, which given with different signal costs.
(1) Relative accuracy (correlation) ranks and cost are shown as the table 1..

Table 1. Level and cost of relative accuracy

relative accuracy range	definition	category	cost
level 1	irrelative or useless	repeated data	0
		overdue useless data	0
		inexistent data	0
		simple literal correlation, but data are not relative	0
level 2	latent relative correlation	a little relative but not full	0.5
		include the data that correlate with range 3 and get by synonymous mean and other means	0.5
level 3	close correlation	full, plenty data	1
		close research and discussion data	1

Relative accuracy coefficient:

$$g(j)=\begin{cases}0 & j\in level1\\ 0.5 & j\in level2\\ 1 & j\in level3\end{cases} \quad (1)$$

(2) Every test event X_i, which has single test sign and test content. The computational method to accuracy ratio $P(Xi)$: select preceding 300 results from every test event, put them divide into 4 groups, J_1= {1，2，… ，40}，J_2= {41，42，… ，100}，J_3= {l01，102，… ，200}，J_4= {201，202，… ，300}, every record j(j=1，2，… ，N), if returns the record number ≥300, the N gets to 300, else returns full record number. Set every group cost:

$$f(j)=\begin{cases}0.45 & j\in J1\\ 0.25 & j\in J2\\ 0.2 & j\in J3\\ 0.1 & j\in J4\end{cases} \quad (2)$$

$$H(X_i)=\sum_{j=1}^{N} f(j)g(j) \qquad (0<N\le 300) \quad (3)$$

$$M(X_i)=\begin{cases}0.45N & 0<N\le 40\\ 0.25N+0.6 & 41<N\le 100\\ 0.2N+1.1 & 101<N\le 200\\ 0.1N+3.1 & 201<N\le 300\end{cases} \quad (4)$$

$$P(X_i)=\frac{H(X_i)}{M(X_i)} \quad (5)$$

If N=0, $P(Xi)=0$.
(3)The calculation comes out:

$$P(A)=\frac{1}{n}\sum_{i=1}^{n}P(X_i) \quad (6)$$

on the basis of the $P(A)$ value may deduce relative accuracy rate. The $P(A)$ value is moved up to a higher position to make known that the mined data is more correlative in the relative accuracy.

Method of survey of cover rate: We suppose A_i act as i time test, $X_j(j=1,2,...,n)$ is the test word in the different test event of different mining.
Get the Matrix:

$$p=\begin{bmatrix}a_{11} & a_{12} & ... & a_{1n}\\ a_{21} & a_{22} & ... & a_{2n}\\ ... & ... & ... & ...\\ a_{m1} & a_{m2} & ... & a_{mn}\end{bmatrix} \quad (7)$$

aij is the returned record at A_i time test that selected j test word.
Establish:

$$a_j^*=Max\{a_{1j},a_{2j},...,a_{mj}\} \quad j=1,2,...,n; \quad (8)$$

$$b_j^*=\min\{a_{1j},a_{2j},...,a_{mj}\} \quad j=1,2,...,n; \quad (9)$$

$$a_i=\sum_j^n(a_j^*-a_{ij}) \quad (i=1,2,...,m) \quad (10)$$

$$b_i=\sum_j^n(a_{ij}-b_j^*) \quad (i=1,2,...,m) \quad (11)$$

The relative cover degree of A_i is:

$$R(A_i)=\frac{b_i}{a_i+b_i} \quad (i=1,2,...,m) \quad (12)$$

From the above, $0\le R(A_i)\le 1$，if results numbers of every word of A_i is the largest，$R(A_i)=1$, Every theme word returns the result is the smallest, $R(A_i)=0$. If $R(A_i)$ is greater, the cover degree is better.

Survey method of Collection velocity: Collection velocity is correlative with the electric network transmission time, the capacity size of subsystem database, performance of system and matching degree. Considering factors such as

electric network and system give excitation, the entire system is excitation driven. Defining the subsystem speed of response as exciter response gene T_g, if the waiting alert time of evaluation system can not receive the respond beyond T_g the instruction sender may shift to a kind of uncertain condition, which may cause the data not to be believed. Therefore, to guarantee the validity is sure that the excitation can be responded. In order to meet this demand, we should let the equation measure up:

$$T_g > T_l + T_c + T_o \tag{13}$$

Here, Tl is the excitation transmission time, Tc is the calculation time of responding excitation, To is the response time of responding excitation. Hence, system dispose accuracy is:

$$\alpha = \max_{k \in \{1,2,\ldots,n\}} \frac{T_{kl} + T_{kc} + T_{ko}}{T_{kg}} \quad 0 < \alpha < 1 \tag{14}$$

Here n is the number of times that subsystem respond in the system operation process, T_{kl} is the k transmission time, and T_{kc} is the calculation time of responding k excitation, T_{ko} is the response time of responding k excitation, T_{kg} is the waiting time for k excitation that set beforehand. If the value of α is smaller, the collection velocity is better at the same T_{kg}.

5. CONCLUDING RELMARKS

The performance test and evaluation of distributed data mining is a complex task, this paper researches mining test model in a distributed condition, and puts forward a test model, methods and indexes of evaluation. In reality, this kind of measurement is a post-treatment event test. For the sake of the raise effectiveness, maybe combine in-treatment event test and other analysis method to analyze distributed data of the system will be better. In this evaluation system, we still need to consider some else problems, for instance, how to balanced the load against every site? How to deal with the correlation of different data structure? How to validly express data and to intercommunicate with person? Wherefore, the parallel of performance test and evaluation of distributed data mining and rating will more be rational and precise.

6. ACKNOLEDGEMENT

I am very pleased to thank Chunhua Ju, Prof, for his instruction.

7. REFERENCES

[1] Y.J. Feng, Z.C. Liu, et al, “Research on Information Retrieval Evaluation Measure System for Search Engine Functions”, Journal of the China Society for Scientific and Technical Information, February 2004,23（1）：63-68.

[2] T. Jiang, B. Zhang, et al, “Design and Implementation of Distributed Network Performance testing system”, Computer Application, January 2005, 25（1）:192-195.

[3] Y.C. Sun, “Temporal Correctness Validation of Distributed Real-time Simulation (DRTS) System”, Journal of System Simulation, July 2005,17（7）: 1553-1555.

[4] L.M. Wang, Y.M. Chai, et al, “Model for Distributed Data Mining Based on Muti-agent”, Computer Engineering and Application, May 2005,40（9）:197-199, 232.

[5] G.F. Tang, B. Qian et al, “The Application of Hybrid Monitor Technology in Performance Monitoring of Parallel and Distributed Systems”, Computer Engineering, July 2005, 17（26）:25-26, 56.

[6] X. Zou, W. Zhang, et al, “Study on Distributed Sequential Pattern Discovery Algorithm”, Journal of Software, July 2005, 17（7）:1262-1269.

[7] Z.Z. Shi, Knowledge Discovery, Beijing: Tsinghua University. Pub, 2002.

Hierarchical Data Clustering Using Ainet Immune Network

Li Liu
School of Information Technology,
Southern Yangtze University
Wuxi, Jiangsu, 214000, China
Email: yxhll@pub.wx.jsinfo.net

ABSTRACT

The clustering problem can be viewed as an optimization problem that locates the optimal centroids of the clusters directly. This view permits the use of evolutionary algorithm to do data clustering tasks. Recently, as a new branch in evolutionary computing, artificial immune systems (AIS) have shown great powerful in knowledge acquisition, pattern recognition, classification, and optimization tasks. The aiNet, one such AIS algorithm exploiting the biologically inspired features of the immune system, performs well on small dimension data clustering tasks. This paper proposes the use of the aiNet to more complex tasks of data clustering. Based on the immune network and affinity maturation principles, the aiNet condensed the raw data and extracted knowledge from the data set. Combined with hierarchical clustering and graph techniques, the immune network showed good performance in data clustering tasks.

Keywords: hierarchical data clustering, artificial immune network, aiNet.

1. INTRODUCTION

Data Clustering is very useful in several exploratory pattern-analysis, grouping, decision-making, and machine-learning situations, including data mining, document retrieval, image segmentation, and pattern classification. Typical pattern clustering activity involves the following steps [1]:
1) Pattern representation (optionally including feature extraction and/or selection),
2) Definition of pattern proximity measure appropriate to the data domain,
3) Clustering or grouping,
4) Data abstraction and
5) Assessment of output.

It is possible to view the clustering problem as an optimization problem that locates the optimal centroids of the clusters directly rather than finding an optimal partition. This view permits the use of evolutionary algorithm to do data clustering tasks. The well-known evolutionary techniques are genetic algorithms (GAs)[2], evolution strategies (ESs), evolutionary programming (EP), and artificial immune network (AIN)[3]. Out of these four approaches, AIN is a novel immune network, with the main features of competitive learning.

In this case, the raw data are the inputs of the immune network. After several generations, the nodes in the network are representative data, reducing redundancy and describing data structure, including their spatial distribution and cluster inter-relations. Artificial Immune Network serves as a pre-processing for the data clustering analysis, compressing data from a given dataset.

The data clustering approach was implemented in association with hierarchical clustering and graph theoretical techniques, and the network performance is illustrated using Iris dataset benchmark dataset.

This paper is organized as follows. Section II introduces artificial immune network, especially aiNet algorithm principles. In section III presents data-clustering procedure among iris dataset, describes the major techniques in use, and discusses parameter setting for aiNet algorithm. Finally, Section IV presents some concluding remarks.

2. ARTIFICIAL IMMUNE NETWORK

2.1Overview of aiNet

The immune network theory, as originally proposed by Jerne [4], hypothesized a novel viewpoint of lymphocyte activities, natural antibody production, pre-immune repertoire selection, tolerance, self/non-self discrimination, memory and the evolution of the immune system. The immune system was formally defined as an enormous and complex network of paratopes that recognize sets of idiotopes, and of idiotopes, that are recognized by sets of paratopes. The relevant events in the immune system are not only the molecules, but also their interactions. The immune cells can respond either positively or negatively to the recognition signal. A positive response would result in cell proliferation, activation and antibody secretion, while a negative response would lead to tolerance and suppression.
There are several immune network models presented in the literature. Most of them are based upon a set of differential equations to describe the dynamics of the network cells and molecules. aiNet is a new immune network model proposed by Leandro Nunes de Castro & Fernando José Von Zuben in 2001[5]. Recent research showed that it had good performance in data analysis. So, it is feasible to choose aiNet model to deal with Hierarchical Data Clustering problems.

The aiNet clusters will serve as internal images (mirrors) responsible for mapping existing clusters in the data set into network clusters. The numbers within the cells indicate their labels, the numbers next to the connections represent their strengths, and dashed lines suggest connections to be pruned, in order to detect clusters and define the final network structure.

Let a shape-space S be a multi-dimensional metric space where each axis stands for a physico-chemical measure characterizing a molecular shape [6]. We will assume a set of unlabeled patterns $X = \{x_1, x_2, \Lambda, x_{N_p}\}$, where each

pattern $x_i, i = 1,\Lambda\ , N_p$ is described by p variables, to characterize a molecular configuration as a point $s \in S$. Hence, a point in S^p specifies the set of features necessary to determine the interactions between $Ab-Ab$ and $Ag-Ab$ that can be mathematically represented as a p-dimensional vector. The possible interactions within the system will be represented in the form of a connectivity graph. Our network model can be formally defined as:

Definition 1: The aiNet is an edge-weighted graph, not necessarily fully connected, composed of a set of nodes, called antibodies, and sets of node pairs called edges with an assigned number called weight, or connection strength, associated with each connected edge.

The following notation will be adopted:

X: data set composed of N_p vectors ($x \in \Re^p$);

C: matrix containing all the N_t network cells ($C \in \Re^{N_t * p}$);

M: matrix of the N memory cells, ($M \subseteq C$);

N_c: Number of clones generated by each stimulated cell;

D: dissimilarity matrix with elements $d_{ij}(Ag-Ab)$;

S: similarity matrix with elements $s_{ij}(Ab-Ab)$;

n: number of highest affinity cells selected to clone and mutate;

z: percentage of the matured cells to be selected;

sd: natural death threshold;

st: suppression threshold.

The aiNet algorithm aims at building a memory set that recognizes and represents the data structural organization. The more specific the antibodies, the less parsimonious the network (low compression rate), whilst the more generalist the antibodies, the more parsimonious the network with relation to the number of antibodies (improved compression rate). The suppression threshold (st) controls the specificity level of the antibodies, the clustering accuracy and network plasticity.

The aiNet-learning algorithm works as follows:

Step 1 at each iteration step, do:

Step 1.1 for each antigen i, do:

Step 1.1.1 determines its affinity d_{ij}, to all the network cells according to a distance metric;

Step 1.1.2 selects the n highest affinity network cells;

Step 1.1.3 clones these n selected cells. The higher the cell affinity, the larger N_c;

Step 1.1.4 Calculate Distance between all N_c cells;

Step 1.1.5 Determine D for these improved cells;

Step 1.1.6 Re-select $z\%$ of the highest affinity cells and create a partial Mp memory cell matrix;

Step 1.1.7 Eliminate those cells whose affinity is inferior to threshold sd, yielding a reduction in the size of the Mp matrix;

Step 1.1.8 Calculate the network $Ab-Ab$ affinity, s_{ij};

Step 1.1.9 Eliminate $s_{ij} < st$;

Step 1.1.10 Concatenate C and Mp, ($C = [c; Mp]$);

Step 1.2 Determine S, and eliminate those cells whose $s_{ij} < st$ (network suppression);

Step 1.3 Replace r% of the worst individuals;

Step 2 Test the stopping criterion.

The network outputs can be taken to be the matrix of memory cells' coordinates (M) and the matrix of intercell affinities (S). While matrix M represents the network internal images of groups of antigens, matrix S is responsible for determining which cells are connected to each other, describing the general network structure.

3. EXPERIMENTS

To illustrate characteristics of data clustering using immune network techniques, we selected aiNet immune network modal to do data clustering among classical benchmark datasets.

3.1 Iris dataset

Iris dataset is perhaps the best-known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances, where each class refers to a type of iris plant. Attribute Information is composed of sepal length in cm, sepal width in cm, petal length in cm, petal width in cm, and class value.

One class is linearly separable from the other two classes, the latter are NOT linearly separable from each other. There are some attributes data overlapping in the other two classes. In this case, 150 input instances are considered as antigens, which input into the aiNet network after normalization. The following fig.1 and fig.2 show attribute data distribution of iris dataset.

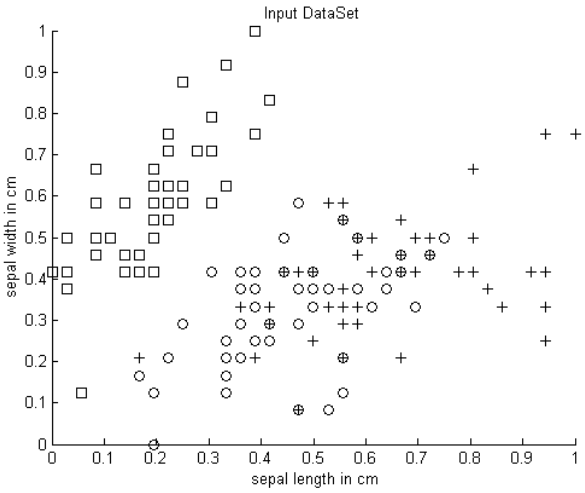


Fig. 1. 1st &2nd dimensions of iris dataset

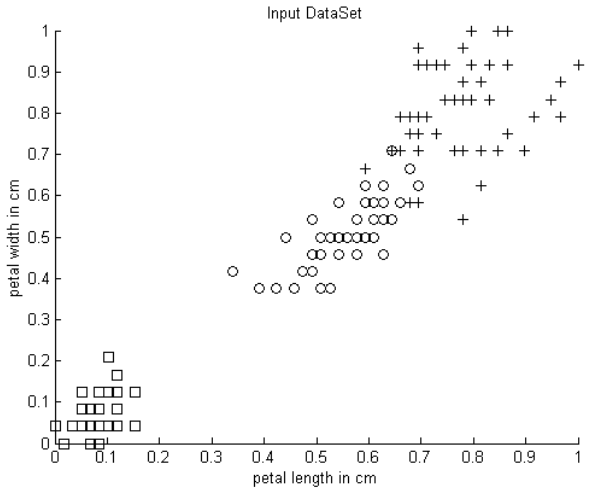


Fig. 2. 3rd &4th dimensions of iris dataset

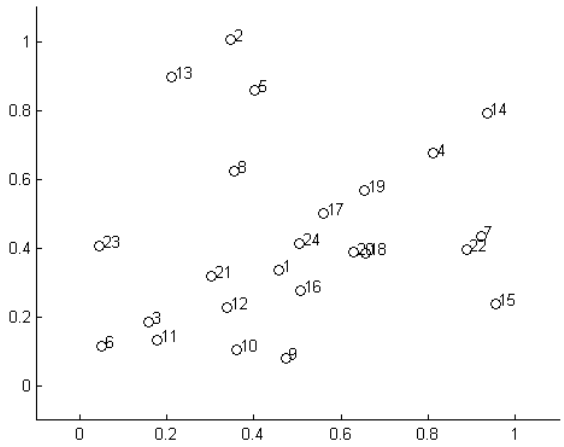


Fig. 3. Two dimensions of network nodes

3.2 Affinity between Ag-Ab, Ab-Ab

In this case, we use Euclidean distance to depict the relationship between Antigen and Antibody or Antibody and Antibody.

$$D(x,y)=\|x-y\|, x,y\in S^L \quad (1)$$

$$Affinity(Ab_i, Ag_j)=1/D(Ab_i, Ag_j) \quad (2)$$

$$Similarity(Ag_j, Ag_k)=D(Ag_j, Ag_k) \quad (3)$$

Output Memory antibodies

The immune network outputs can be taken to be the matrix of memory antibodies $Ab\{m\}$ and their matrix of $affinity(S)$. $Ab\{m\}$ Corresponds to the set of network nodes and $affinity(S)$ are responsible for concentration of the edges of the network.

In the experiments, after 25 iteration, the resultant memory matrix was composed of m=24 antibodies, corresponding to a compression rate $CR=84\%$, reducing the problem to 16% of its original complexity (size).

While $L>3(L=4)$, visualizing the antigens is a difficult task. Fig.3 only illustrated the 1st &2nd dimensional data of memory antibodies. So we need to take advantages of several hierarchical clustering techniques to interpret the generated network.

3.3 Hierarchical Clustering and dendrogram

A dendrogram is defined as a rooted weighted tree where all terminal nodes are at the same distance (path length) from the root [8]. For the purpose of getting the clustering number and the distribution of representative data (memory antibodies), three characteristics can adequately describe a dendrogram: its topology, labels and clustering heights. Fig.4 illustrates the dendrogram representation for the iris memory data clustering.

Hierarchical methods can produce a series of solutions ranging from m clusters to a solution with only one cluster present. In this case, the large variations in heights allow us to distinguish 3 clusters among the network antibodies, which is consistent with iris data cluster number.

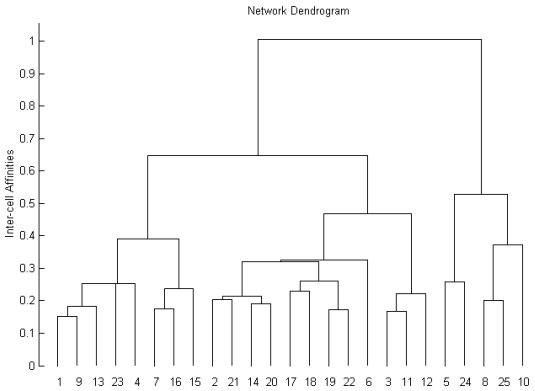


Fig. 4. Dendrogram of network nodes (*ts=0.15 M=25*)

3.4 Parameter setting

To apply the aiNet to data clustering problem, a number of parameters have to be defined. In this case, ts=0.15, td=1.0,n=4,gen=25,z=10. Fig.5 illustrated the network building procedure. The net size reduced from 107 to 24 nodes. Figure8 illustrated the network building procedure. The net size reduced from 44 to 13 nodes.

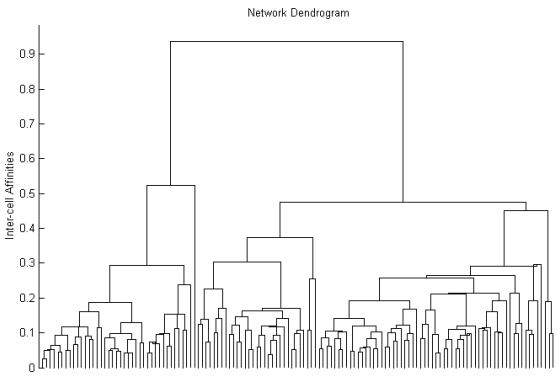


Fig. 5. Dendrogram of network nodes (*ts=0.01 m=142*)

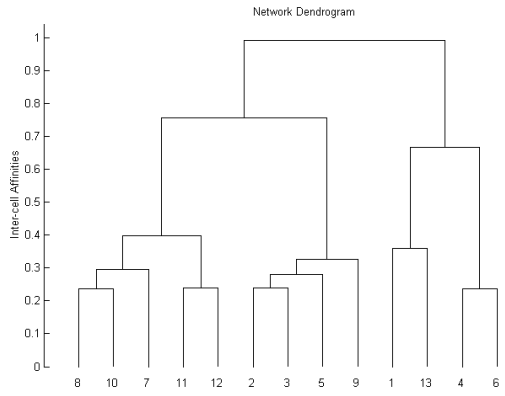


Fig. 6. Dendrogram of network nodes (*ts=0.2, m=13*)

In particular, *ts* is a critical parameter for aiNet. The larger values of *ts*, the more memory antibodies of the network. So *ts* control the compression rate and convergence of the network. Figure9 showed trade-off between number of output networks nodes (M) and suppression threshold (*ts*).

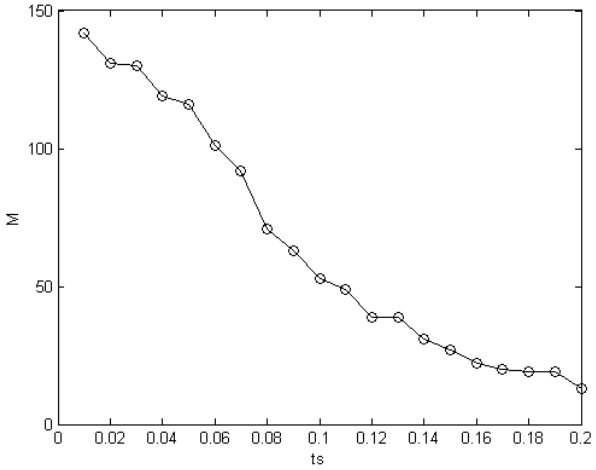


Fig. 7. Trade-off between *M* and *ts*

4. SUMMARY AND CONCLUSIONS

First, we can see that in the presence of three distinct clusters of memory antibodies, each of which with different number of antibodies, connections and strengths. These clusters map those of the original data set. Second, the number of antibodies in the network is much smaller than the number of data samples, characterizing architecture suitable for data compression. Finally, the shape of the spatial distribution of antibodies follows the shape of the antigenic spatial distribution.

5. REFERENCES

[1] A.K. JAIN, R.C. Dubes, “Algorithms for Clustering Data,” Prentice-Hall advanced reference series, Inc., Upper Saddle River, 1988.

[2] D.E. Goldberg, ” Genetic Algorithms in search, Optimization and Machine Learning”, Addison-Wesley publishing Co., Inc., Redwood City, CA, 1989.

[3] N.D.C. Leandro, T. Jon, “Title of paper optional Artificial Immune Systems: A Novel Computational Intelligence Approach”, Springer-Verlag, 2002.

[4] N.K. Jerne,”Towards a Network Theory of the Immune System”, Ann. Immuno, (Inst. Pasteur) 125C, 1974, 373-389.

[5] L.N. Castro, Z.B. von, ”aiNet: An artificial immune network for data analysis”, USA:Idea Group Publishing, 2001,231-259.

[6] A. S. Perelsen, G. F. Oster, “Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-Nonself Discrimination”, J. theor. Biol., 81, 1979,pp. 645-670.

[7] T. Na, V.R. Vemuri, ”An Artificial Immune System Approach to Document Clustering”, SAC’05, March 13–17, 2005, Santa Fe, New Mexico, USA.

[8] F.J. Lapointe, Legendre,” The Generation of Random Ultrametric Matrices Representing Dendrograms”, Journal of Classification, 8, 1991, 177-200.

A CONCEPTUAL MODEL FOR CONSTRUCTION OF DOMAIN ONTOLOGY

Qing Yang, Hongxing Liu
School of Computer Science and Technology, Wuhan University of Technology
Wuhan, Hubei 430063, China
Email: qingyang@public.wh.hb.cn liuhongxing@mail.whut.edu.cn
Kecheng Liu
Informatics Research Center, the University of Reading
Reading, Berkshire RG6 6AH, U.K.
Email:k.liu@reading.ac.uk

ABSTRACT

The construction of domain ontology is regarded as a process of modeling in this research. The modeling of ontology conceptual model (OCM) is an important approach to identify the concepts and establish the relations of concepts. The essential primitives of OCM are discussed in detail, and the emphasis is put on the analysis of semantic relationships. The basic semantic relationships, which should be expressed in OCM, are: is-a, subset, aggregation, instance-of, association; then, based on semantic analysis, the modeling of OCM is illustrated by an example. The research in this paper sets the foundation for further development of methodology.

Keywords: Ontology engineering, Ontology methodology, Conceptual modeling, Semantic analysis.

1. INTRODUCTION

Ontology first became a popular research topic in the Knowledge Engineering community, and it is now being widely considered valuable for other research areas (e.g. Web-Services). The reason is largely due to the initiative of Ontology, namely, establishing the shared and common understanding of some domain that can be communicated across people and computers; in other word, ontology is probably the practical way to support the web-wide interoperability in semantic layer. In this paper, *domain ontology* specially refers to the category, which captures and expresses the knowledge valid for a particular type of domain [11].

Since 1990s, many ontology languages and development tools are already available; some ontology has been put into use [1]. But the methodologies for building ontology are relatively immature, because there is not widely accepted methodology so far; ontology development still largely depends on the experience of developers. Which kinds of primitive should be expressed in ontology? How many *semantic relationships* should be or could be expressed? What is the essence of constrain and how to expressed it? There are still some fundamental problems unsolved.

The researchers and practitioners in software engineering community have had the consensus, the development of software is an evolving process, and the most important activities involved are construction and transformation of models. As a kind of software or the part of software, it is a practical choice for ontology development to follow the successful route.

Ontology development should follow the strictly defined process. According to the definition in [4], development-oriented activities include specification, conceptualization, formalization and implementation.

This research focuses on the conceptualization of ontology. We present the concept of Ontology Conceptual Model (OCM), such an OCM is the basis of semantic analysis approach for construction of domain ontology, and a bridge to realize the transformation from domain knowledge to specific knowledge model.

The remainder will unfold in the following manner. Section 2 presents the architecture for knowledge representation and in detail discusses the primitives which should be expressed in ontology conceptual model; the emphasis is put on the analysis of semantic relationships between concepts. Section 3 presents a practical semantic analysis approach for the modeling of OCM by means of a simplified example. The last section concludes with an indication of further work.

2. THE ESSENTIAL PRIMITIVES IN ONTOLOGY CONCEPTUAL MODEL

Some literature adopts different sets of terms to express their understanding on ontology. Generally, the terms used in these literatures includes: concept, class, attribute, property, relationship, axiom, constrain, instance, etc. But some terms and definitions are ambiguous and even conflicting. The confusion is partly due to that the terminology is actually related to multi-layers of cognition and abstraction. Fig. 1. describes the architecture for knowledge representation and transformation, in which knowledge is identified and expressed in three layers (i.e. requirement layer, conceptual layer, and representation layer).

2.1 Requirement Layer

Goldstein et al define things and properties as follows: The real world is composed of things having properties that are inherent and exist objectively whether or not they are observed or recorded [5]. A property can be characterized as descriptive or associative according to its semantic role: a descriptive property expresses some characteristic of a thing; an associative property relates two or more things.

We notice that the associative property is separately defined. Associative property expresses the relation between things. For example, each student should be registered to a

program, so thing *student* has an associative property registered-program; registered-program establishes the relation between thing student and thing program.

To some extent, associative property describes dynamic characteristic of the thing, while descriptive property represents the static characteristic of the thing.

2.2 Representation Layer

In Fig. 1., term '*ontology*' is assigned a specific meaning. It refers to the model, which the concepts and their attributes, as well as the semantic relationships between the concepts, are organized in the form of a computable knowledge model. To a great extent, ontology could be regarded as a knowledge-representation system or the part of knowledge-representation system, and is dependent of the knowledge model.

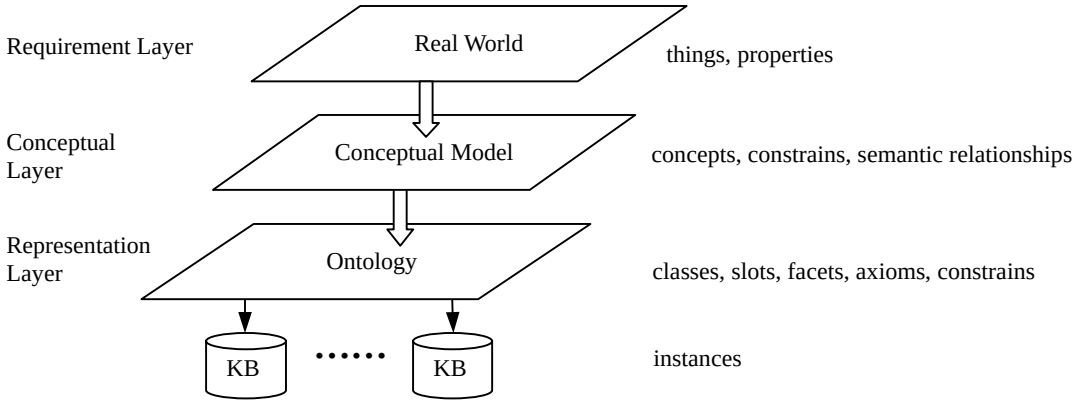


Fig. 1. The layers of knowledge representation

Different ontology development environment provides support to different knowledge model and has relating terms. The knowledge model of Protégé is frame-based; frames are the principal building blocks of the knowledge base [8]. We take terms of Protégé as example in representation layer.

Protégé ontology consists of classes, slots, and facets. A class is the ontology representation of a concept in a domain; slots describe attributes of class; facets describe properties of the slot (e.g. type, cardinality). Protégé ontology implicitly supports multiple inheritances; one class can have more than one superclass. Protégé also supports mataclass architecture, which enables users to adapt the knowledge model to meet the extra requirement of their domain and task. Protégé knowledge base is consisted of instances of classes with specific values for each slot.

Ontology is a formal, explicit specification of a shared conceptualization [11], and it may take a variety of forms. Which kinds of primitives the ontology should express in order to necessarily meet the fundamental features (i.e. 'conceptualization', 'explicit', 'formal', 'shared')? It is what we will discuss in section 2.3.

Some approach, such as database system also build domain model by using entity, attributes and relations, but a database model (e.g. relational data model) is impossible to explicitly express semantic relationships between data and the constrains on data within database itself because of its inherent representational limitations.

In some previous practice, ontology is simply considered as taxonomy, represents concepts and 'is-a' relationship between concepts. Obviously ontology will at least include taxonomies of some domain, but should differ from taxonomy by revealing richer semantic relationships among concepts.

2.3 Conceptual Layer

Ontology conceptual model (OCM) aims at capturing and representing the domain knowledge in abstract and visual way. It is independent of specific knowledge model and development environment, and is the common understanding shared by domain experts and ontology developers.

We will respectively discuss three kinds of primitives, which should be expressed in OCM.

1) Concepts and Attributes

A concept is the abstract representation of a 'thing' in the real world. An attribute is the abstract representation of property. For example, concept S*tudent* has attributes {*Name*, *Library no*, *Registered-time*, *Registered-program*}. The first three attributes are the abstract representation of descriptive property; the last attribute is the abstract representation of associative property, it establishes a kind of relation between concept *student* and concept *program* (we will discuss it in next part).

The distinction between concept and attribute is not absolute. A concept has its own attribute, and meanwhile it (or its instance) may be the attribute of another concept.

2) Constrains

Ontology represents common understanding of domain knowledge, which is shared and communicated by a group of people, so the group of people should have consensus on constrain (or definition) of concepts.

Semiotics, the theory of signs, is to some extent relevant to the research of Ontology. Liu K. analyses the concept of norms in [7]. Norms are developed through practical experience of agents in a society, and in turn have functions of directing, coordinating and controlling actions within the society. The agent refers to an individual person, a group of people, even a community or society. According to how norms control human behavior, norms are further classified as five types, which are perceptual norms, cognitive norms, evaluative norms, behavioral norms and denotative norms.

Constrains of concept in ontology mostly express the cognitive norms, that is, describes how the group of people

interpret the information they received. Denotative norms should also be taken into account when constrains are defined. Denotative norms direct the choices of signs for signifying, and are culture-dependent. For example, 'supervisor' is a term popularly used in British university, whereas 'adviser' is more popular in North America. 'Supervisor' and 'adviser' almost have the same meaning in the context of university, so they should be expressed as synonym in 'University' ontology.

3) Semantic Relationships

Semantic relationships are the abstraction of the inherent relations among things and properties. Database community has made much effort to identify and represent semantic relationship. Veda C. Storey summarizes and presents taxonomy of semantic relationships in [10]. Robert et al. analyze the most common abstractions (namely, inclusion, aggregation and association) in [5]. The semantic relationship existing in real world is universal, semantic analysis is also the basis of ontology design. Applying data abstraction theory to analyze concepts and the relations among concepts, we could identify five basic semantic relationships in the context of ontology.

Is-a

The relationship 'is-a' is the most important inclusion abstraction. *Ca* is-a *Cb*, both *Ca* and *Cb* is concept, *Ca* is specialization of *Cb*. *Ca* is relatively a generic concept, *Cb* is a specific concept. For example, graduate is a Student, Undergraduate is a Student too; both Graduate and Undergraduate are specialization of Student.

The most important feature of 'is-a' is inheritance: *Ca* is-a *Cb*, *Ca* thus inherits all the attributes of *Cb*. Multiple inheritance is permitted in OCM: *Ca* is-a *Cb*, *Cb* is-a *Cc*, *Ca* thus inherits the attributes from both *Cb* and *Cc*.

Another important feature of 'is-a' is transitivity: *Ca* is-a *Cb*, *Cb* is-a *Cc*, then, *Ca* is-a *Cc*.

Subset hierarchy

When a generic concept is defined as the union of non-overlapping specific concepts, the specific concepts form a partition of the generic concept; the generic concept is called generalization of specific concepts. For example, *Student* is generalization of graduate and undergraduate, so graduate and undergraduate is a partition of Student.

It is necessary to examine all the partitions relating to a generic concept. For example, Student is the generalization of Undergraduate, Graduate, Part-time, Full-time, Oversea and Home. Three groups, undergraduate and Graduate, Part-time and Full-time, Oversea and Home, form different partition of Student. Clearly, overlap exists between these three groups; a subset hierarchy is thus formed.

Aggregation (and component-of)

Aggregation is an abstraction in which a relation between concepts is considered as a higher level concept. For example, Program is an aggregation of name, affiliated-subject, degree, duration, description and steering committee; conversely, name, affiliated-subject, degree, duration, description, steering-committee are respectively 'component-of' Program.

If *A* is 'component-of' *B*, *A* will be regarded as the attribute of *B*. Some attributes, as we mentioned above, are the abstract representation of associative property, and establishes the relations between concepts. Such attributes are called composite attribute. For example, affiliated-subject relates Program and Subject, and indicates which subject the program is affiliated with, so affiliated-subject is a composite attribute.

Instance-of

Instance-of could be regarded as another kind of inclusion abstraction between a composite attribute and the instance of a concept. Instance-of is thus called instantiation in some literatures.

Ca is instance-of *Cb*, means that the type of *Ca* is *Cb* and the occurrence of *Ca* should be an instance of *Cb*. For example, concept Student has attributes {name, library-no, registered-time, registered-program}. Registered-program is a composite attribute, each occurrence of registered-program is an instance of Program, and so registered-program is instance-of Program.

Association (and member-of)

Association is an abstraction in which a collection of the instance of a concept is considered as another (higher) concept. *Ca* is an association of (the instances of) *Cb*; conversely (an instance of) *Cb* is member-of *Ca*.

For example, concept Subject has attributes {name, description, programmes, steering committee}, composite attribute steering committee is an association of concept Academic, (an instance of) Academic is member-of steering-committee. In this case, 'association' establishes a relation between a composite attribute and a concept. Another example, concept Council is an association of concept Academic, the semantic relationship 'association' occurs between two concepts.

We should notice that the relationship 'association' defined above is different from the 'association' defined in UML.

In the ER model, an attribute is the abstract representation of descriptive property; a relation between entities expresses an associative property; whereas an associative property is expressed as a composite attribute in OCM. Relationship along with entity and attribute is the primitive in ER model, represents and establishes the relation between entities. The initiative of database system is to store data and to provide support for transaction processing, ER model do meet the requirements of modeling operational data in conceptual layer. But ontology focuses on representing and storing knowledge, the concepts and attributes are organized in more explicit and relevant form, thus it is possible to express richer semantics in ontology. The methodology of database design has had profound influence on conceptual modeling, so we should notice the differences when conducting ontology design.

3. The SEMANTIC ANALYSIS APPROACH FOR CONCEPTUAL MODELLING OF DOMAIN ONTOLOGY

It is clear that the relations of concepts are identified and

established by analyzing semantic relationships, according to the discussion above. In this section, we will illustrate the process of modeling OCM by means of a simplified example; the related activities in the modeling process will be conducted based on the identification and expression of semantic relationships. An OCM is eventually established by constructing a set of taxonomies and identifying the relations between taxonomies.

3.1 Group Concepts by Identifying 'Aggregation'

Part of concepts elicited from domain 'British University' is listed in concept table as follows. For example, *Student* is aggregation of name, library-no, registered-time, and registered-program, so they are grouped together. Composite attribute is underlined so as to be distinguished from descriptive attribute.

Student {name, library-no, registered-time,
registered-program},
Undergraduate, Graduate, Master, PhD, Part-time,
Full-time, Oversea, Home,
Program {name, affiliate-subject, degree, duration,
description, director}
Subject {name, steering-committee}
Staff, Administration, Technical-support, Academic

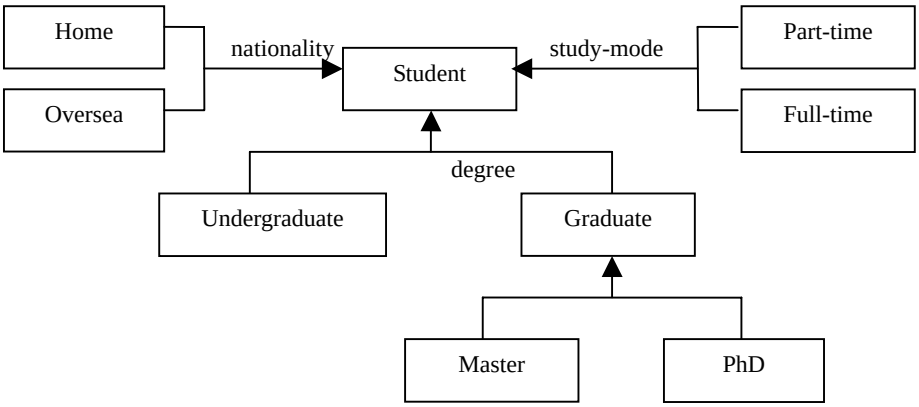


Fig. 2. Expression of 'Student' taxonomy

3.2 Identify 'Is-a' And 'Subset'

We realize that there are three generalization sets for Student by analyzing concept table, each set is a partition of Student and indicated by a role (i.e. degree, or nationality, or study-mode). Three-generalization sets overlap, so a Student could be classified via each of these sets (e.g. someone can be oversea full-time master student).

Fig. 2. Shows a generalization hierarchy existing in domain 'British University', that is, taxonomy Student. Single arrow indicates 'is-a' relation between concepts, much detail is omitted in order to highlight the semantic relationships 'is-a' and 'subset'.

3.3 Identify 'Instance-of' And 'Association'

As we discussed in section 2, each composite attribute establishes a relation between two concepts. Domain ontology usually includes a set of taxonomy; the relations between taxonomies are established by identifying 'instance-of' and 'member-of'. For example, director is component-of Program and instance-of Academic, steering-committee is component-of Subject and member-of Academic, so a link between taxonomy Program and Academic is established by director, and a link between taxonomy Subject and Academic is established by steering-committee.

4. CONCLUSIONS

In this paper, the construction of domain ontology is considered as a modeling process: ontology conceptual model (OCM) is first constructed, and then OCM will be transformed to some knowledge model supported by an ontology development environment.

The emphasis of this paper is to define the primitives which should be expressed in OCM. The essential primitives include: concepts and attributes, constrains, and semantic relationships; the basic semantic relationships are: is-a, subset, aggregation (and component-of), instance-of, and association (and member-of). We further illustrate the modeling of OCM by a simplified example, which shows that the analysis and identification of semantic relationships is the central activities in the process of OCM modeling.

We intend to extend this research in two ways. Firstly, we will investigate the transformation from ODM to specific knowledge model; the emphasis will be put on the transformation to semantic-web based knowledge model and language (that is, RDFS and OWL). Secondly, based on the previous research, we are going to analyze and identify the domain-dependent semantic relationships, and build ontology for some specific domain (for example, software engineering).

5. REFERENCES

[1] O. Corcho, M. Fernández-López, et al, "Methodologies, tool, and languages for building ontologies. Where is their meeting point", Data & Knowledge Engineering, 2003, Vol.46: 41-64.

[2] Z.H. Deng, S.W. Tang et al, "Overview of Ontology Research", Journal of Beijing University (Natural Science), 2002, 38(5): 731-738.

[3] M. Fernández-López, "Overview of methodologies for building ontologies", Proceeding of the IJCAI-99 orkshop on Ontologies and Problem-Solving Methods

(KRR5). http://sunsite.informatik.rwth-aachen.de/CEUR-WS/Vol-18/

[4] M. Fernández-López, A. Gómez-Pérez et al, "Building a chemical ontology using METHONTOLOGY and the ontology design environment", IEEE Intelligent System & their application, 1999, 4(1): 37-46.

[5] R. C. Goldstein, V. C Storey, "Data Abstraction: Why and how?", Data & Knowledge Engineering, 1999, Vol.29: 293-311.

[6] N. Guarino, C. Welty, "Ontological analysis of taxonomic relationships", Proceedings of ER2000: Conference on Conceptual Modeling. Lecture Notes in Computer Science, Springer-Verlag, 2000, 210-224.

[7] K. C. Liu, Semiotics in Information System Engineering (1st Ed), Cambridge: Cambridge University Press, 2000.

[8] N. F. Noy, R. W. Fergerson, et al, "The Knowledge model of Protege-2000: Combining interoperability and flexibility", 12th International conference in Knowledge Engineering and Knowledge Management. Lecture Notes in Artificial Intelligence, Springer-Verlag, 2000, 17-32.

[9] N. F. Noy, D. L. McGuinness, "Ontology development 101: A guide to creating your first Ontology", http://protege.stanford.edu/publications/ontology_development

[10] V. C. Storey, "Understanding Semantic Relationship", Vary Large Data Base (VLDB) Journal, 1993, 2(4): 455-488.

[11] R. Studer, V. R. Benjamims, et al, "Knowledge engineering: Principles and methods", Data & Knowledge Engineering, 1998, Vol.25: 161-197.

Fast computing answer to query streams With wavelet deconstruction*

Xin Chen, Weixing Chen
Department of Computer Information, Beijing Information Technology Institute
Beijing 100101, China
Jianqiang Niu
School of Electronic and Informational Engineering, Henan University of Science and Technology
Luoyang, 471039, China
Shuo Yang
School of Computer Science, the University of Manchester
Manchester, M139pl, UK

ABSTRACT

Continuous and real-time data streams exist in large-scale database systems and Internet applications. Fast query and analysis are often desirable in order to efficiently process the streams. In this paper, we first investigate the aggregate computation in stream queries by analysing the features of and the relation between aggregate query language and aggregate query rewriting. Based on the analysis, we present an efficient fast query computational approach, in which equivalent subset synopsis is designed based on wavelet approach. Finally, by deploying an implementation of the fast query model in a network monitoring and analysis system, we test and verify the efficiency of aggregate computation on stream queries. The result shows that the wavelet-based approach has distinct advantage to traditional method.

Keywords: query streams; aggregate computing; subset synopsis; wavelet approach

1. INTRODUCTION

The research and application of query streams are an extension of data query based on materialized aggregate view and the integration with communication and statistical research. Stream query distinctively concentrates on computation and analysis on current 'flowing' streams stored in an infinite storage space. Such query on fast, consecutive and large volume of streams has become one of the core functionalities in many application systems. For example in telecommunication and IP network, performance data of distributed components need to be continuously monitored, collected, queried and analysed in order to report real time system status, detect faults and recover from failures. Other examples include transaction information in chain retailing industries and credit card tractions in financial services etc.

Computation and storage of consecutive streams are usually expensive for the Database Management System (DBMS). Therefore, a focused processing on current 'interesting' streams is desirable to reduce the cost of computational and storage operations. Some applications need to process prioritized transaction that could be potentially beneficial for decision-making and prediction. Research in consecutive stream query and its algorithm design has been a hotspot in data query with comprehensive value in applications.

In this paper, we first investigate the aggregate computation in stream queries and analyse the features of and the relation between aggregate query language and aggregate query rewriting. Secondly, by focusing on the mathematical features of consecutive and instantaneous stream query we present a fast and efficient computational approach that concentrates on streams in large-scale network applications. In the approach, an equivalent subset synopsis is designed based on wavelet approach. Equivalent subset synopsis data have the basic features of large datasets, and the computational results of both are of equivalence. Finally, we implement and deploy our approach in an application environment to execute effective and efficient aggregate data queries. The results are also analysed and compared with conventional approaches.

2. AGGREGATE LANGUAGE ANDREWRITING

A. Aggregate Query Language

In most cases, stream data queries focus on aggregate computation, namely aggregate data query, which is the primary and key query in stream query computation. The basic functions of aggregate data query are Count, Sum, Multiply, Average, Percentage, Max and Min etc.

Definition 1 Let s= (T_1,…, T_n) be a stream data set，which is a finite set of relation symbols and its components. An Instance I of s is a mapping from relation symbol to respective component.

Let $\Re = (R, \leq, +, \times, 0, 1)$ be a Real Field Structure, in which domain R is a set of real number, = and $\leq$ are predictions, $+$ are $\times$ functions, 0 and 1 are constants. The first order logic of the structure is noted FO_R. If we extend the structure to aggregate operations, we can generate an Aggregate Operation Language FO_R^{agg} [1]. Let $\varphi(x, \bar{y})$ and $\psi(x, \bar{y}, z)$ be two formulas, in which $x, \bar{y}, z$ are free variables, then *Count* $C_x \varphi(x, \bar{y})$ defines a local function on a property $\bar{y}$, in which the domain of

property $\overline{y}$ is $\{\overline{y} \mid \exists x \varphi(x, \overline{y})\}$. The function defines the corresponding relations between a tuple $\overline{a}$ and cardinal number $C_{\overline{a}}$ of set $\{x \mid \varphi(x, \overline{a})\}$.

Property $\overline{y}$ is $\{\overline{y} \mid \exists x, z \psi(x, \overline{y}, z)\}$. The function defines the corresponding relations between a tuple $\overline{a}$ and Sum $S_{\overline{a}}$ of elements in multi-set $\{\{x \mid \psi(x, \overline{a}, z)\}\}$.

Multiply $\Pi_x^z \psi(x, \overline{y}, z)$ defines a local function on a property $\overline{y}$, in which the domain of property $\overline{y}$ is $\{\overline{y} \mid \exists x, z \psi(x, \overline{y}, z)\}$. The function defines the corresponding relationship between a tuple $\overline{a}$ and Product $p_{\overline{a}}$ of elements in multi-set $\{\{x \mid \psi(x, \overline{a}, z)\}\}$.

The definitions of average, percentage, max and min can be derived from above definitions. We can mark the difference between formula arguments and function variable using a headline. Meanwhile, function variable corresponds to formula free variable without restriction of aggregation, in the form of $\overline{C_x \varphi(x, \overline{y})}(\overline{y})$. For example, for a SQL query with aggregate function, its aggregate query language FO_R^{agg} is:{[y，s]|s=$\overline{\Sigma_x^z (R(x, y, z))}(y)$}.

B. Aggregate View and Query Rewriting

In the processing of aggregate data, the main challenge is the computation on Aggregate View [2]. Especially in an application environment such as Data Warehouse and OLAP, the view upon which query is performed is usually materialized as an aggregate view. The main reasons are that: (1) it is impossible or expensive to perform direct operation; (2) it is not necessary to draw data from source dataset for every single operation. Materialized View is usually used to provide users the objects that aggregate queries perform on.

Materialized Aggregate View is the view generated in advance. A query on Materialized Aggregate View is therefore the process of query rewriting corresponding to certain Materialized Aggregate View or a Rule in a database.

Aggregate query can be achieved by two types of predictions in Database relationship set R and View set V. We extend database D to database D_V, in which every View $v \in V$ can be explained as a Relation v^D. If q is a query on a view in V, we call q^{D_V} is a relation derived from value assignment of query q in D_V.

Definition 2[3] Let q, q' be two aggregate queries in view $R \cup V$. If for all databases D, $q^{D_V} = q'^{D_V}$, we call q and q' are model V equivalent, noted as $q \equiv_v q'$.

Sum $\Sigma_x^z \psi(x, \overline{y}, z)$ defines a local function on a property $\overline{y}$, in which the domain of

Definition 3[3] Let q be a query and q' a query on View set V of relation set R and $R \cup V$. If $q \equiv_v q'$, we call q' is a rewriting of q using V. If q' is a rewriting of q using V, then we call q' is a partial rewriting of q with V. If q' contains at least one atomic view in V, we call q' is a complete rewriting of q with V. For a query q on database pattern S, and a set of views V on S, if there exists a q', a rewriting with V, we call q is direct computable from q'.

The definitions explain query computing on View. Thus they also address aggregate query processing on aggregate views. Generally, there are two steps in a query rewriting. Firstly, all possible query rewriting candidate plans are produced. Secondly, comparison and verification of candidate rewriting plans take place in order to search for an optimal plan.

Query rewriting is performed by query rewriting algorithm. A rewriting algorithm is used to generate a rewriting query q' from a view set V and a query q in a database scheme S. Given an algorithm, if there exists a rewriting generated from view set V, we note $V \vdash q$. If for any view set V and any query q, $V \vdash q \Rightarrow V \models q$, we call rewriting algorithm is determinative. If for any view set V and any query q, $V \models q \Rightarrow V \vdash q$, then we call the rewriting algorithm is complete.

3. AN EFFICIENT APPROACH TO QUERY STREAMS COMPUTATION

The running status of a network system can be characterized by the criteria of streams in the network. One of the features of stream query is that it only requires results with an allowed approximation rather than accurate results. Thus it is possible to timely return the query results within a very short period of time for performance analysis and diagnosis.

There are various methods for approximate queries. Different approximate methods can present different approximate probabilities and approximate characteristics. A typical data stream query has the following form:

```
SELECT   count (*)
FROM     student, teacher, score
WHERE    student.cno=teacher.cno=score.cno
```

In many applications, it is desirable to return query results as efficient as possible. Fast query has its advantage in aggregate query that it can provide efficient response to queries on important streams. Generally, there are two categories of fast query

methods - online query processing method and pre-computed synopsis method. The former is controlled and optimized by users. The latter builds and stores synopsis before queries and returns the result on query. The synopsis method requires synopsis update and maintenance, which are based on pre-processing.

enerally, a stream pre-processing algorithm is required to be capable of abstracting the basic features of synopsis subset of streams. The efficiency of synopsis retrieval is provided by a cache with limited storage space. The accuracy of synopsis is guaranteed by a reasonably approximated value range.

This type of approximated online query processing is beneficial for most application environments in which large amount of streams exist. For example the pattern analysis and fault detection in telecommunication networks are used to detect unusual pattern and usage based on approximated queries rather than accurate data.

The aggregate computational model of streams is presented in Fig. 1..

Let Q (D_1, D2…Di) be a query on D_1, D2…Di, where D_i is a tuple of data. After query rewriting, the computation will operate on streams within the fast query-computing engine. The streams at this stage are of simultaneous feature and the data in the streams are not reduplicative. Therefore at this phase, the query can only represent a collective view of the data. After obtaining the intermediate results from the engine, a certain algorithm ensuring probability and effectiveness will perform maintenance on the intermediate results before returning final approximated results by rewriting mechanism.

The core of the model is the fast query-computing engine, in which synopsis data subset plays an essential role. By the feature abstraction in the engine, each stream D_i becomes abstract synopsis data subset Record unit$_i$ noted as $S(D_i)$. The restrictions on $S(D_i)$ are that (1) the synopsis subset $S(D_i)$ is smaller than the tuple in data stream D_i in terms of scale. The scale of $S(D_i)$ is of one or more logarithm of $|D_i|$. $S(D_i)$ is an abstract subset of D_i; (2) the tuple in a data stream can arrive in random order. At any time, the fast query model can provide an approximated result by combining synopsis data subsets $S(D_1), S(D_2), \cdots, S(D_n)$

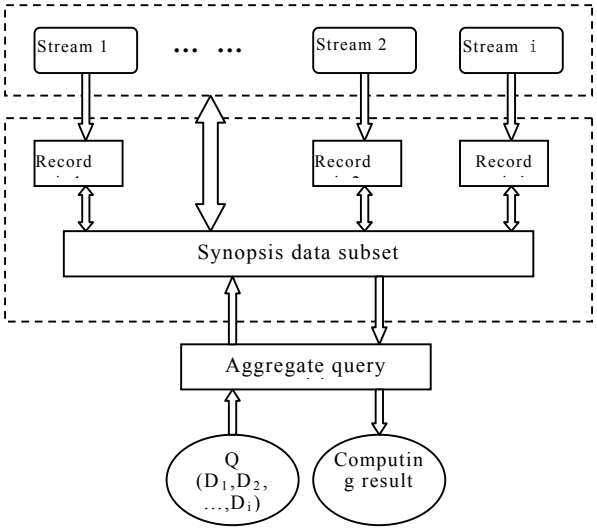


Fig. 1. An efficient approach to query streams computation

Thus the approximation can effectively improve computational efficiency. There are many approaches to implementing the approximation, such as histograms, Fourier transform, Bayesian method, and wavelet transform.

4. WAVELET CONSTRUCTION OF SYNOPSIS SUBSET IN FAST AGGREGATE QUERY MODE

The core functionality of the fast aggregate query model is the maintenance and combination of synopsis subsets. Basically there are two approaches for querying synopsis subsets: single-scale and multi-scale approaches. The single-scale approach can employ histogram, sampling or wavelet method. However those methods are not isolated from each other and can be combined in certain circumstances to construct an optimized synopsis for query. We present a single-scale approach by the Haar wavelet method in order to achieve more effective query synopsis.

The presentation of functions in wavelet analysis is usually used as a type of analytical tools for hierarchical deconstruction. By discrete coefficients, partial and holistic features of a function can be presented and analysed so that the characteristic of momentary signal mutation can be well captured. Data streams also have such characteristic of momentary signal mutation and wavelet can be used to construct the synopsis subset in the engine.

Definition 4 There is a function $\psi \in L^2(R) \bigcap L^1(R)$, and $\psi(0) = 0$. ψ is extended and translated into a group of functions:

$$\psi_{a,b}(x) = |a|^{-1/2} \psi(\frac{x-b}{a}), (a,b \in R, a \neq 0) \quad (1)$$

We call $\{\psi_{a,b}\}$ is analytical wavelet, and ψ is basic wavelet, where a is a spread fact，b is a smooth

fact ◦ The Haar wavelet function $\psi(x)$ is a basic wavelet:

$$\psi(x) = \begin{cases} 1, & 0 \le x < 1/2 \\ -1, & 1/2 \le x \le 1 \end{cases} \qquad (2)$$

Let the parameters a,b in (1) be discrete value, so that a continuous wavelet is transformed into a discrete wavelet. We have the formula:

$$\psi_{m,n}(x) = a_0^{m/2}\psi(a_0^m x - nb_0), (m,n \in Z)$$

In certain application environments, the group of wavelet functions $\{\psi_{m,n}, m,n \in Z\}$ is interdependent rather than in quadrature. Let $\psi, a_0 = 2, b_0 = 1$ in order to let $\{\psi_{m,n}, m,n \in Z\}$ be orthonormal basis. The simplest orthonormal basis is the Haar wavelet basis. The feature of Haar wavelet basis is that it is simple for understanding and implementation. Therefore fast query synopsis in the engine is constructed based on Haar wavelet basis. In detail, it is formed by deconstruction and reconstruction by Haar wavelet. General hierarchical deconstruction of Haar wavelet can be described by tree structure. Single-scale Haar wavelet method uses recursion on source data to calculate average values in pairs and coefficients of each scales, as listed in Table 1..

From the sequence in Table 1., we reconstruct the Haar wavelet hierarchical deconstruction, the error tree, as shown in Fig. 2..

In the error tree, the nodes are Haar wavelet coefficients and the leaves are source data. The Haar wavelet-based synopsis on distributed source data is suitable in fast query answering systems in that the coefficients of the tree structure can be maintained and updated dynamically.

We implemented the computational model and deployed it in an application environment of BEA Web logic 7.0 application servers and Oracle 9i database server in a Sun Solaris 8 operating system. The Spool command is sent via SQL Plus tool in Oracle 9i database to execute a query. Text results are used to generate data mart from pipes. A snapshot is shown in Fig. 3..

Table 1. Coefficients of haar wavelet destruction

Hierarchy	Average value	Coefficients
4	[9,5,6,8,6,4,3,3,5,7,4, 8,4,6,7,3]	… …
3	[7, 7, 5, 3, 6, 6, 5, 5]	[2,-1,1,0,-1,-2,-1,2]
2	[7, 4, 6, 5]	[0,1,0,0]
1	[5.5, 5.5]	[1.5,0.5]
0	[5.5]	[0]

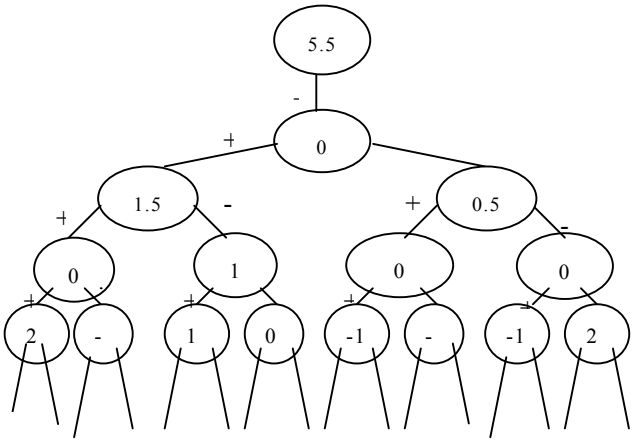


Fig. 2. Error tree of wavelet hierarchical deconstruction

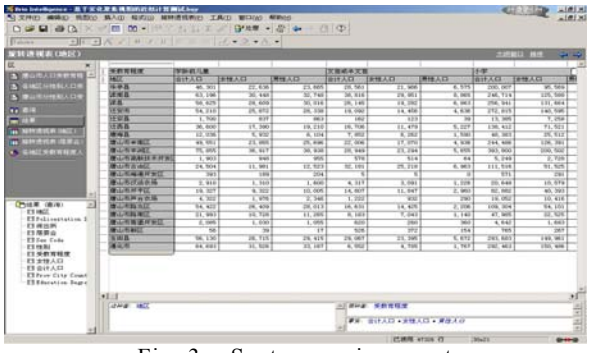


Fig. 3. System environment

The efficiency of query and system performance are improved within allowable error rate, as listed in Table 2. and shown in Fig. 4..

Table 2. Comparison of performances

Data Scale (Number of people)	Time of conventional query (second)	Time of fast query (second)	Error rate
36098107	432.328	390.142	11.1%
191731667	439.737	393.575	12.7%
204452343	441.953	401.768	10.4%

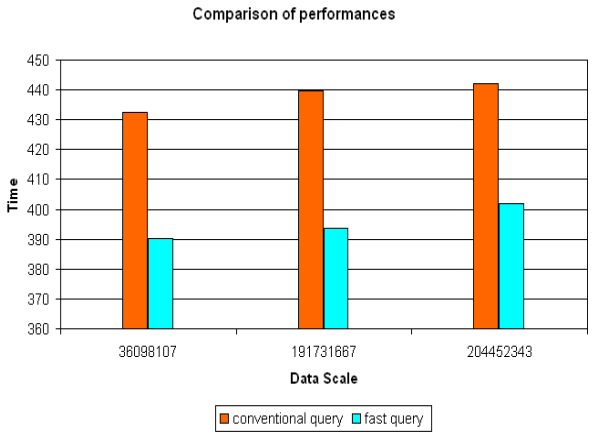


Fig. 4. Comparisons of Performances

The results indicate that by the fast query approach, query time is less than average of conventional query time, which means the efficiency of query processing on streams is significantly improved.

5. CONCLUSION

Stream query processing has a distinct feature that queries usually operate on continuous, instantaneous and 'flowing' data stored in a limited space. By analysing the characteristic of aggregate queries of streams, we investigate the relation between aggregate query language and aggregate query rewriting. Further, in order to take advantage of these features, we present a fast computational approach that processes consecutive and real time streams in network systems. The approach employs aggregate equivalent subset synopsis based on wavelet methods. Finally, by implementing the model and testing it in an application environment with different scales of datasets, we compare the results and analyse the improvement of efficiency of the fast aggregate query provided by the new approach. The result indicates that average query time is 83.92% of query time by conventional approach, which is a significant improvement of system performance.

6. REFERENCE

[1] S. Grumbach, M. Rafanelli, et al, Querying Aggregate Data [A]. Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database System[C]. New York, NY, USA: ACM, 1999, 174-184.

[2] S. Grumbach, L. Tininini, "On the content of materialized aggregate views," [A]. Proceedings of the Nineteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database System[C]. New York, NY, USA: ACM, 2000, 47-57.

[3] S. Cohen, W. Nutt, et al, Algorithms for Rewriting Aggregate Queries Using Views [A]. Pro. Symposium on Advances in Databases and Information Systems[C].Prague,Czech Republic: ADBIS-DASFAA,2000,65-78.

[4] S. Abiteboul, O.M. Duschka, Complexity of Answering Queries Using Materialized Views [A]. Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database System[C]. New York, NY, USA: ACM, 1998, 254-263.

[5] S. Chaudhuri, G. Das, et al, "A Robust Optimization-Based Approach for Approximate Answering of Aggregate Queries [A]", New York, NY, USA: ACM SIGMOD, 2001, 295-306.

[6] N. Alon, Y.Matias, et al, "The space complexity of approximating the frequency moments," [J]. Journal of Computer and System Sciences, 1999, 58(1): 137-147.

[7] G. Cormode, P.Indyk, et al, "Fast mining of tabular data via approximate distance computations," [A]. IEEE ICDE[C]. San Jose, California, USA: IEEE, 2002, 605-616.

[8] C. Aggarwal, J.Han, et al, "A framework for clustering evolving data streams," In VLDB, 2003.

[9] D. Kifer, S. Ben-David, et al, Detecting Chang in Data Stream., Toronto, Canada: In VLDB, 2004.

[10] S. Schmidt, H. Berthold, et al. Qstream: Deterministic Querying of Data Streams, Toronto, Canada: In VLDB, 2004.

[11] Query Languages and Data Models for Database Sequences and Data Streams, Toronto, Canada: In VLDB, 2004.

Voting Model Based Rank Fusion Algorithms for Metasearch

Yao Yu, Zhu Shanfeng*, Chen Gang, Chen Xinmeng
Computer School, Wuhan University, Wuhan Hubei 430072, China
Email: yaoyu928@163.com
***Institute for Chemical Research, Kyoto University, Uji Kyoto 611-0011, Japan**

ABSTRACT

The rank fusion problem for metasearch engines was studied based on voting model in this work. In addition to two widely discussed classical voting rules: Borda and Condorcet, some elimination voting algorithms and their variants, including Kemeny and Nanson methods, were analyzed through a graph theoretic approach. As Kemeny ranking problem was NP-hard complexity, a new heuristic elimination voting algorithm was proposed. Some experiments were carried out on TREC data for evaluating these voting algorithms on rank fusion. The experiments indicate that these elimination algorithms have comparable performance with Borda algorithms, and sometimes outperform it.

Keywords: Rank Fusion, Elimination Voting, Voting Model, Metasearch, Information Retrieval.

1. INTRODUCTION

1.1 Background

In the recent years, the category of work known as metasearch has been concerned extensively, and there has been much research done in this field. In the context of the World Wide Web, metasearch is the process of retrieving and combining information from multiple sources that index effectively common data sets, the Web [1]. Metasearch has been developed to overcome the shortcoming of single independent search engine and to improve the retrieval effectiveness. As we know, the search result of one search engine according to a query is typically a ranking list of documents relevant to the given query. Thus, in metasearch work, the problem of rank fusion is to compute a "consensus" ranking list by fusing several result lists returned from multiple search engines.

Several rank fusion methods have been proposed in the literature. A major distinction between the methods is that they can be classified based on whether:

(i) They rely on the rank;
(ii) They rely on the score;
(iii) They require training data or not [2].

Some experiments indicate: when different systems are commensurable, combination using similarity scores is better than combination using only ranks; but when multiple systems have incompatible scores, a combination method based on ranked outputs rather than the scores directly is the proper method for combination [3,4]. And in most cases, only the ranks of the documents in each search engine are available. In this work, we will focus on rank-based rank fusion methods.

1.2 Related Work and Motivations

Metasearch problem can be modeled as a election voting procedure in which we only need rank information. We can deem the source search engines as voters and all ranked documents as candidates, then metasearch problem is actually to select a voting procedure to make collective choices on these candidates. Voting procedures can be traced back to two centuries ago in Western Europe where elections for parliament and administrative were popular. Two famous voting algorithms, Borda rule and Condorcet rule, are proposed to deal with sophisticated cases. In addition to these two algorithms, other voting extensions such as Black procedure and Kemeny [5,6], have been proposed and widely employed in the elections. Aslam and Montague [1,7] adopted Borda rule and Condorcet rule to build data fusion algorithm and carried out the experiments on the data in TREC3, TREC5 and TREC9. They found their algorithms usually outperform the best-input system and are competitive with existing metasearch strategies. Dwork et al. studied Kemeny method for metasearch on the Web, and found that the corresponding problem of computing optimal Kemeny ranking is NP-hard for given k (k>3) full (or partial) rankings on N candidates [8].

In the rest of this paper, we first present the voting model and some basic elimination voting algorithms for rank fusion in the next section. And we propose a new heuristic algorithm based on elimination voting model in section 3. In section 4, these voting algorithms and their combination are compared in the fusion experiment on TREC data. It shows that these elimination voting algorithms achieve comparable performance with other voting algorithms such as Borda.

2. VOTING MODEL AND ALGORITHMS

In this section, we present the voting model and some basic ideas of corresponding Graph Theory algorithms for rank fusion.

2.1 Voting Model

Given a set of n candidates (i.e. n documents of search result) N={1, 2, …, n}, a ranking τ with respect to N is a permutation of all elements of N which represents a voter's (i.e. search engine's) preference on these candidates. For each $i \in N$, $\tau(i)$ denotes the position of the element i in ranking τ, and for any two elements $i, j \in N$, $\tau(i) < \tau(j)$ implies that i is ranked higher than j by the ranking τ. There is a collection of k rankings $\tau_1, \tau_2, \cdots, \tau_k$, which are provided by a set of voters K={1, 2, …, k} respectively. The voting based rank fusion algorithm is to aggregate these k rankings into a "consensus" ranking π on these n candidates. Since plurality rule cannot handle the election with more than two candidates, Borda and Condorcet voting model have been proposed. Based on these two fundamental algorithms, many other voting extensions are introduced with the development of democracy, such as Kemeny, Nanson, and Baldwin. One type of voting algorithms is called as elimination voting in which the whole voting procedure is divided into multiple stages, and one or more candidates are eliminated in each stage until the winner is

determined. Since most of these voting algorithms are proposed for single-seat winner election in the beginning, we need transform them into a full ranking on the candidates. Many voting algorithms can be explained in a graph theoretic approach. For the above voting (rank fusion) problem, we illustrate some typical approaches and corresponding graph theoretic algorithms in the subsections.

2.2 Borda Rule

It employs a point system to measure the difference among candidates. For each voter, the lowest candidate will be assigned a nonnegative integer score *x*. Then the other candidates above will be scored with an interval of positive integer *y*, such as *x*+*y*, *x*+2*y* ... *x* +(*n* - 1) *y*. In a typical case, we give *x* = 1, *y* = 1. Then considering any two candidates *i*, *j*∈*N* in the aggregate ranking π, π(*i*) < π(*j*) implies that:

$$\sum_{l=1}^{k}(n+1-\tau_l(i)) > \sum_{l=1}^{k}(n+1-\tau_l(j)) \cdot \quad \textbf{Eq.(1)}$$

Construct a digraph *G* = (*N*, *E*) in which the set of vertices corresponds to the set of candidates *N*, and for any pair of vertices *x* and *y*, create an arc (*x*, *y*) with the weight $\omega(x,y) = \left|\{l \in K, \tau_l(x) < \tau_l(y)\}\right|$. The indegree of *x* is:

$$Indegree(x) = \sum \omega(m,x),\ m \in N \quad \textbf{Eq.(2)}$$

The Borda algorithm is actually to find ascending sorted list of vertices *N* with respect to indegree.

2.3 Condorcet Rule

It considers relative majority between each pair of candidates, and proposes that the candidate defeating any other candidates in pairwise comparison should be elected as winner: *x* is the winner of election if for any other candidate $i \in N, \left|\{l \in K : \tau_l(x) < \tau_l(i)\}\right| > \left|\{l \in K : \tau_l(i) < \tau_l(x)\}\right|$. However Condorcet winner may not exist, since it may have voting cycles (paradox) in the candidates. Montague et al. extended it to a full ranking by a kind of QucickSort algorithm [7]. The candidates in different voting cycles can be ranked according to Condorcet rule, and the candidates in same cycles will be ranked arbitrarily, which is called as extended Condorcet criterion (ECC)[9]. Dwork et al. also introduced a procedure, which is called as Local Kemenization, to make the final result conforming to ECC [8].

In a graph *G* = (*N*, *E*), the set of vertices represents the set of candidates *N*, and an arc (*x*, *y*) between vertices x and y represents $\left|\{l \in K : \tau_l(x) < \tau_l(y)\}\right| \geq \left|\{l \in K : \tau_l(y) < \tau_l(x)\}\right|$. This graph consists of several Strongly Connected Components (SCC), which identify different voting cycles. It is similar to tournament graph. Then the ranking with respect to Condorcet rule is actually to find a Hamiltonian path in the graph. Montague et al. named this graph as Condorcet graph and proved it contains a Hamiltonian path, which can be obtained through an efficient Quicksort procedure [7].

2.4 Kemeny Rule

This algorithm is based on the concept of Kendall-τ distance between two rankings, which counts the total number of pairs of candidates that are assigned to different relative orders in these two rankings. In other words,the Kendall-τ distance between two rankings τ_1, τ_2 is defined as:

$$D(\tau_1,\tau_2) = \left|\{(i,j) : \tau_1(i) < \tau_1(j) \wedge \tau_2(j) < \tau_2(i)\}\right| \quad \textbf{Eq.(3)}$$

Kemeny ranking is an optimal ranking π with respect to the given *k* rankings $\tau_1, \tau_2, \cdots, \tau_k$, which are provided by the voters and can minimize the total Kendall-τ distance:

$$D(\pi;\tau_1,\tau_2,\cdots,\tau_k) = \sum_{i=1}^{k} D(\pi,\tau_i) \quad \textbf{Eq.(4)}$$

The complexity of finding a Kemeny optimal ranking is a NP-hard problem even for *k* >3 [8].

In Kemeny approach the graph *G* is constructed in the same way as Borda voting. An optimal Kemeny ranking is actually to find a directed acyclic subgraph of *G* with maximum sum weight of all arcs.

2.5 Nanson and Baldwin Rule

These two voting algorithms belong to elimination voting, and come from Borda. In Nanson voting proposed by Edward John Nanson, Borda score will be calculated in each stage [10]. The candidates failing to reach average Borda scores of all alternatives will be eliminated. The procedure will be repeated until the winner is chosen. Nanson voting combines the merits of both Borda and Condorcet, and if a Condorcet winner exists, it will choose the winner. It was adopted by two universities in Australia, University of Melbourne and University of Adelaide for the election of Council members [11]. Baldwin voting is actually a variant of Nanson voting [12]. In each stage of Baldwin voting, only the candidate with the lowest Borda score will be eliminated.

In these two approaches, the graph *G* is constructed in the same way as Borda voting. Compared with other voting algorithms, Nanson and Baldwin belong to elimination voting in which the whole voting procedure consists of multiple stages. In each stage of Nanson voting, the vertices of which the Indegree is greater then average Indegree of all vertices will be eliminated. Then a new graph will be constructed and this procedure will be repeated until the last vertex is left. Baldwin voting is similar to Nanson voting. In each stage of Nanson voting, the vertex of which has the greatest Indegree out of all vertices will be eliminated in each stage

3. NEW ELIMINATION ALGORITHMS

The basic idea of elimination voting is to choose the winner by remove the losers in multistage procedures. In each stage, remove part of candidates, which are obviously ranked lower by most of voters. The procedure will be repeated until the winner is chosen. Elimination voting can be formulated in a multistage graph algorithm in which one or more vertices with large Indegree will be eliminated until the last vertex left. The vertex with large Indegree indicates it should be ranked lower in merged list with high probability. For obtaining a merged ranking in rank aggregation, we can also first eliminate “best” candidate, or eliminate the candidate that may be “worst” or “best”, but has the largest difference with other candidates. From the graph theoretic point of view, this means removing the vertex according to its *Indegree* or *Outdegree*.

Based on the graph formulation for voting, we propose several new elimination algorithms that can be divided into five categories. The construction of the graph is same with Borda.

3.1 MaxIn

In this algorithm, the vertex with maximum Indegree and its incident arcs will be removed. The candidate with respect to this vertex will be ranked in the lowest not filled position in merged list. The procedure will be repeated until only one vertex left, which will fill the highest position of the merged list.

3.2 MinIn

In this algorithm, the vertex with minimum Indegree and its incident arcs will be removed. The candidate with respect to this vertex will be ranked in the highest not filled position in merged list. The procedure will be repeated until only one vertex left, which will fill the lowest position of the merged list. Due to limited space, we don't present MaxIn and MinIn algorithms here.

3.3 MaxDiff

This algorithm, shown in Table 1., incorporates both Indegree and Outdegree of the vertex. In each stage, we compute the absolute value for each vertex the difference between the Indegree and Outdegree. The vertex with the maximum absolute value of difference will be eliminated, that is, eliminate x with $max_{i \in N}\{|Indegree(i)\text{-}Outdegree(i)|\}$. Note that $Indegree(i)$ and $Outdegree(i)$ stand for the strength of voters putting candidate i in lowest position and the highest position, respectively. After finding the vertex, if $Outdegree(i)$ is larger than $Indegree(i)$, corresponding candidate will be put at the highest not filled position in merged list. Otherwise, it will be put at the lowest not filled position in merged list. The procedure will be repeated until we rank all the candidates.

This algorithm is a heuristic algorithm for computing a full ranked list of all candidates. The construction of the graph G takes time $O(kn^2)$. And computation of γ takes time $O(n^2)$. Thus, the total computation takes time $O(kn^2)$.

3.4 MinOut & MaxOut

For any two candidates x and y, if all the k rankings $\tau_1, \tau_2, \cdots, \tau_k$ give different ranks to x and y, that is, either $\tau_i(x) > \tau_i(y)$ or $\tau_i(x) < \tau_i(y)$, $i \in K$. Then in the graph $G=(N, E)$, we can get for all $x \in N$, $Indegree(x) + Outdegree(x) = k$. Under this condition, MinOut is equivalent to MaxIn, and MaxOut is also equivalent to MinIn. Thus, we only need analyze three of them MaxIn, MinIn and MaxDiff.

Table 1. MaxDiff algorithm

MaxDiff Algorithm
01: $u \leftarrow 1$ and $v \leftarrow n$.
02: Let graph $G = (N, E: \omega)$
03: corresponding to k input rankings on N.
04: $\gamma = max_{i \in N}\{|Indegree(i)\text{-}Outdegree(i)|\}$.
05: denote i^* the vertex with the largest γ.
06: If $Indegree(i^*) \leq Outdegree(i^*)$
07: $\pi(i^*) \leftarrow u$,
08: $u \leftarrow u + 1$;
09: Else
10: $\pi(i^*) \leftarrow v$,
11: $v \leftarrow v - 1$.
12: Construct a new Graph G by removing i^* from N
13: and incident arcs from E.
14: If $v \geq u$
15: goto Line 04;
16: Else
17: exit and output the ranking π.

4. EXPERIMENTS

In this section we will conduct some experiments to compare the performance of these voting algorithms by using the retrieval systems participating in Web track of TREC [1] as the input for data fusion. For judging the efficiency of retrieval, two common evaluation measures are applied: Average Precision and Precision at 10(P@10).

4.1 Experiment Data

TREC is an annually held conference since 1992, which provides test-bed and benchmarks for retrieval systems on large document collections. A set of sample information needs is designed as topics including title, description and a short narration. For each topic, each participant is required to submit top 1000 ranked documents. The relevance of the documents is judged by the reviewers who originally construct these queries or by the evaluation software.

In our experiment, we make use of searching results provided by retrieval system attending Web track of TREC9 and TREC2001. There are two categories of retrieval systems participating in Web Track of TREC9, one is the group of retrieval systems searching the documents by only considering the title of the topic, the other group considers much more information such as description, manually constructed queries. On the other hand, in Web Track of TREC2001, it only includes retrieval systems of title-only category.

Since retrieval systems come from same company will decrease the performance of the Condorcet algorithm [7], we select top 10 independent retrieval systems in the Web Track of TREC2001 and TREC9 as input with respect to average precision. The selected retrieval systems in TREC9 are iit00m, jscbt9wll2, ric9dpn, Nenm, acsys9mw0, hum9tdn, pir0Watd, tnout9f1, NRKprf20 and Sab9web3, and in TREC2001 those are fub01be2, JuruFull, ricMM, jscbtawt14, Lemur, ok10wt3, hum01tlx, msrcn1, tnout10t2 and iit01tfc.

4.2 Evaluation Measures

In our experiments, two evaluation measures are used: Average Precision, Precision at Top 10 results (P@10). Since the measure of average precision, which incorporates both precision and recalls criteria, can reward the system retrieving relevant documents quickly, it is widely used to measure the performance of retrieval system. Average precision is computed in this way: when each relevant document occurs, precision value is computed. Then average precision is the average of all these values.

According to several studies [13,14], people seldom go beyond top 10 hits of the result, which means the list at the top is the most important to the users. Consider this point, another measure, precision at Top 10 results (P@10), is applied.

Given searching results of a retrieval system, these two measures can be easily obtained by a program "trec_eval" provided by TREC. As there are usually 50 topics in each track of TREC competition task, mean average precision and P@10 for all 50 topics are reported.

4.3 Procedure

In this work, we discuss four voting algorithms: Borda, MinIn, MaxIn and MaxDiff. The ranking with respect to Condorcet rule (ECC) can be guaranteed by Local Kemenization procedure. That is, after achieving the initial merged ranking, we can carry out Local Kemenization procedure on initial ranking to make it to fulfill ECC. Thus,

[1] http://trec.nist.gov

there are 8 voting algorithms or their combinations for rank fusion in our experiment. For each TREC data, there are total 9 runs in the experiment, having from 2 to 9 input retrieval systems respectively. All 50 topics in TREC2001 and TREC9 web track are examined in the experiment. According to algorithms discussed in last section, we compute the top 1000 merged documents for each topic. Given these results, the program "trec_eval" will be executed to obtain performance information such as mean average precision and mean P@10 for all 50 topics in the track.

In our experiment, we focus on following questions:

1) Are these voting algorithms effective for rank fusion?

2) Does Local Kemenization procedure (Extended Condorecet Criterion) carried out on initial ranking increase the performance of retrieval?

3) The comparison of performance among voting algorithms.

4.4 Experimental Results

The summary of experimental results is shown in Table 2.. Since each voting algorithm will be executed in all 9 runs as input retrieval systems increasing, we compute the average performance in all 9 runs. From Table 2., we can easily see that all voting algorithms outperform input system in average, which means voting algorithms for rank fusion are practicable and effective for improve the performance of retrieval in our experiments. In Table 2., "LK" stands for Local Kemenization procedure carried out on initial merged ranking.

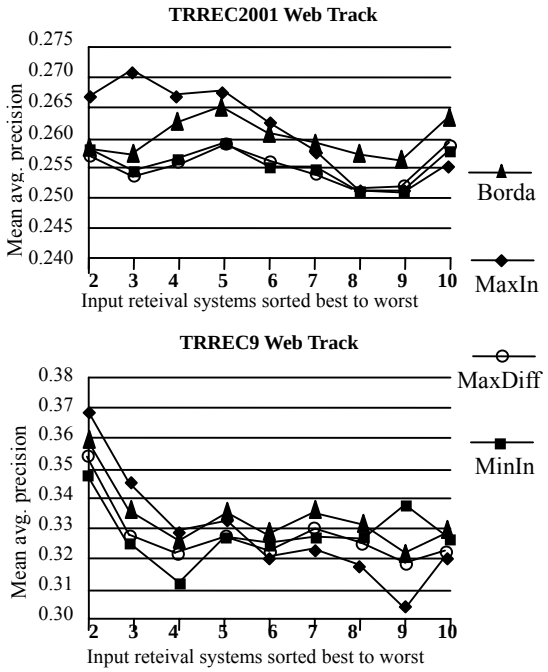


Fig. 1. Experiment on elimination voting for rank fusion

We can compare the performance of voting algorithms with and without LK procedure to identify its effectiveness. For data set from TREC9, we can observe obvious gain for all voting algorithms on Average Precision, and also some improvements on P@10. On the other hand, for data set from TREC2001, almost no improvement can be discerned on Average Precision, and even worse the performance of P@10 is decreased. We think this may due to the characteristic of input retrieval system, such as relative low average precision of retrieval systems participating in Web track of TREC2001.

For comparing the performance of each voting algorithm, we can refer to Table 2. and Fig. 1.. In Fig. 1., we present the mean average precision of merged results by different voting algorithms as input retrieval systems increase. We notice that these four voting algorithms have small difference in performance. In the experiment of TREC2001 data set, the voting algorithms ranked descending with respect to mean average precision are MaxIn, Borda, MinIn and MaxDiff. In the experiment of TREC9, the voting algorithms are ranked as Borda, MaxIn, MinIn and MaxDiff. For the measure of P@10, MinIn has the best performance and MaxIn is the worst for the data set of TREC2001. While in the experiment of data set of TREC9, MaxIn has the highest P@10, and Borda is the worst one.

5. CONCLUSIONS

In this work, we study the rank fusion problem via voting algorithms. In addition to two widely discussed classical voting rules: Borda and Condorcet, some elimination voting algorithms and their variants are analyzed in a graph theoretic approach. As Kemeny ranking problem is NP-hard, a new heuristic elimination voting algorithm which is straightforward and easy for implementation is proposed. Some experiments have been carried out on TREC data for evaluating these voting algorithms on rank fusion. In the experiment, we find that these elimination algorithms have comparable performance with Borda algorithms, and sometimes outperform it. Future works include many more experiments on randomly selecting input retrieval systems and weighted version of voting algorithms for rank fusion.

REFERENCES

[1] A.A. Javed, M. Mark, "Models for Metasearch", Proceedings of ACM SIGIR 2001, USA, 2001, 276-284.

[2] M.E. Renda, U. Straccia, "Web metasearch: rank vs.score based rank aggregation methods",Proceedings of the 2003 ACM symposium on Applied computing, USA,2003, 841- 846.

[3] N.J. Belkin, P.B. Kantor, et al, "Combining evidence of multiple query representation for information retrieval", Information Processing & Management, 1995,31(3): 431-448.

[4] D.F. Hsu, I. Taksa, "Comparing Rank and Score Combination Methods for Data Fusion in Information Retrieval",Information Retrieval,Springer Netherlands, 2005, 8(3):449-480.

[5] Black Duncan, The Theory of Committees and Elections, Springer Press, 1998.

[6] J.G. Kemeny, Mathematics Without Numbers, Daedalus 88, 1959.

[7] M. Mark, A. A. Javed, "Condorcet Fusion for Improved Retrieval", CIKM2002, USA, 2002, 538-548.

[8] R. Dwork, R. Kumar, et al, "Rank aggregation methods for the web", WWW10, USA, 2001,613-622.

[9] M. Truchon, "An extension of the condorcet criterion and kemeny orders", cahier 98-15 du centre de Recherche en Economie et Finance Appliquees, 1998.

[10] J.H. Humphreys, A.B. Poland, "Proportional

Representation A Study In Methods Of Election", Kessinger Publishing, Whitefish, USA, 2004.
[11] I. McLean, "Australian electoral reform and two concepts of representation", APSA Jubilee Conference, Australia, 2002.
[12] I. McLean, "Nanson:Social Choice and Electoral Reform", Australian Journal of Political Science, 1996, 31(3):369-386.
[13] Jansen B.J., Spink A., et al, "Real life, real users, and real needs: a study and analysis of user queries on the web", Information Processing and Management, 2000, 36(2): 207-227.
[14] C. Silverstein, M. Henzinger, et al, "Analysis of a very large altavista query log", Technical Report SRC 1998-014,Digital Systems Research Center, 1998.

Table 2. The Performance of Voting Algorithms for Rank Fusion in the Experiments

DATA SET	RANK FUSION ALGORITHMS	MEAN AVG. PRECISION				P@10			
		Min	Max	Avg.	St.dev	Min	Max	Avg.	St.dev
TREC9	Input System	0.2159	0.3519	0.2498	0.0398	0.290	0.518	0.3552	0.0602
TREC9	Borda	0.3207	0.3587	0.3323	0.0105	0.438	0.466	0.4460	0.0094
TREC9	Borda+LK	0.3338	0.3654	0.3523	0.0087	0.442	0.466	0.4571	0.0087
TREC9	MaxIn	0.3024	0.3688	0.3284	0.0179	0.432	0.492	0.4556	0.0176
TREC9	MaxIn+LK	0.3342	0.3688	0.3478	0.0109	0.436	0.492	0.4578	0.0148
TREC9	MinIn	0.3117	0.3477	0.3281	0.0092	0.438	0.468	0.4496	0.0100
TREC9	MinIn+LK	0.3216	0.3549	0.3424	0.0087	0.442	0.468	0.4571	0.0092
TREC9	MaxDiff	0.3174	0.3551	0.3279	0.0103	0.432	0.470	0.4580	0.0107
TREC9	MaxDiff+LK	0.3304	0.3604	0.3478	0.0081	0.442	0.470	0.4569	0.0077
TREC2001	Input System	0.1890	0.2226	0.2006	0.0104	0.278	0.362	0.3220	0.0253
TREC2001	Borda	0.2561	0.2650	0.2603	0.0029	0.380	0.412	0.3976	0.0091
TREC2001	Borda+LK	0.2444	0.2695	0.2589	0.0073	0.378	0.402	0.3938	0.0083
TREC2001	Maxin	0.2509	0.2710	0.2614	0.0061	0.376	0.392	0.3842	0.0054
TREC2001	Maxin+LK	0.2431	0.2662	0.2557	0.0742	0.372	0.392	0.3813	0.0063
TREC2001	MinIn	0.2512	0.2592	0.2555	0.0030	0.388	0.412	0.3998	0.0076
TREC2001	MinIn+LK	0.2415	0.2655	0.2558	0.0073	0.378	0.406	0.3953	0.0094
TREC2001	MaxDiff	0.2508	0.2595	0.2554	0.0031	0.386	0.412	0.3969	0.0075
TREC2001	MaxDiff+LK	0.2419	0.2666	0.2562	0.0072	0.382	0.404	0.3951	0.0071

Data Mining in Grid environment Using GT4 and OGSA-DAI Technology

Xiufeng Jiang
College of Mathematics and Computer Science
Fuzhou University, Fuzhou, Fujian, China
Email: jxf1963@21cn.com
Jie Li
College of Mathematics and Computer Science
Fuzhou University, Fuzhou, Fujian, China
Email: mhxy-lijie@163.com

ABSTRACT

The characteristic of grid is that various resources are distributed and dynamically changed, so it is necessary to research data mining system in grid environment. In this paper we first introduce grid and data mining concepts in grid environments. Secondly we bring forward five-level architecture of GT4 and OGSA-DAI-based data mining, and then we discuss the design of data mining grid portal, and the implementation pattern of centralized data mining service and parallel data mining service. Finally discuss the implementation pattern of OGSA-DAI-based data integration service.

Keywords: data mining, Grid Computing, distributed data integration, data Grid, OGSA-DAI, WSRF.

1. IMPORTANT INFORMATION

The important characteristics of grid are distribution and dynamic. Various resources of grid are non-concentrative, but distributed on geographically different places, and these resources are also dynamically changed. Just these characteristics of grid makes grid data mining system not be restricted on the traditional centralized system yet, but migrate to grid system[1]. As for grid environment, most of traditional data mining systems are based on GT2 or GT3, less on GT4. The difference between GT3 and GT4 is that GT4 is based on WSRF and GT3 is based on OGSI. This difference makes it difficult to migrate a service from GT3 environment to GT4 environment[2].

2. The architecture of grid data mining system

This system is built with layered thinking. It is divided into five layer: end user, data mining grid portal, data mining and data integrate service, the basic service of GT4, distributed data and compute resource. The architecture is showed as figure 1. In this figure, at first service(service agent), thirdly service agent invoke data mining service, fourthly data mining service invoke data integration service and various standard grid services to finish data mining task. The data mining service is built on GT4, and the data integrate service is based on OGSA-DAI.

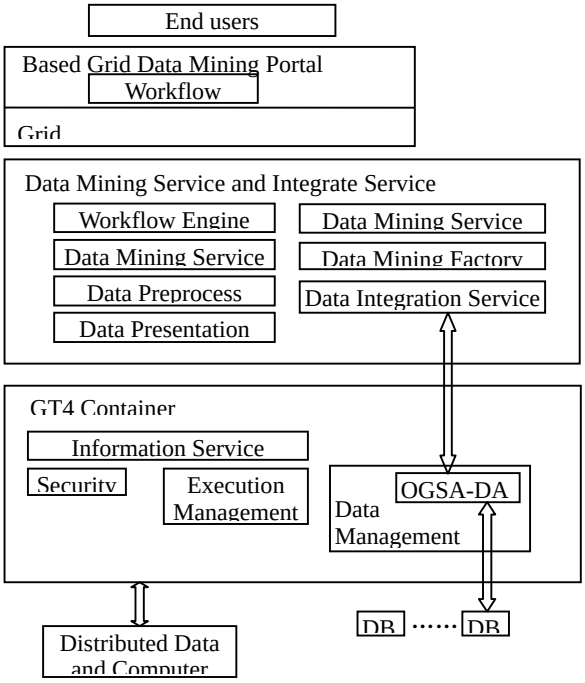


Fig. 1. Data Mining in Grid environment Using GT4 and OGSA-DAI architecture

2.1 Grid service level

Grid service level provides standard grid service interface. It includes monitoring and discovery, security, data management and execution management.

2.2 Grid service layer

Data grid service layer implements grid data mining system services, including the following various services:

1) Data mining service factory[3]: It is a specialized, persistent service that is responsible for creating instances of Grid data Mining services and returns Endpoint Reference of the instance. It is created at startup time of the GT4 container and registers himself automatically within the registry.

2) Data mining service registry: It is a specialized, persistent service that is derived from the standard OGSA registry service. It is created at startup time of the OGSA

This work was partially supported by Grid Portal Industry Technology Research and Development Project, under Grant Fujian Technical Committee and Data Interface of Scientific Computing in Grid Environment, under Grant Fujian Education Committee.

container. A client application is able to browse or search the registry for interesting services. Then the service registry returns Endpoint Reference of service as an answer to queries.

3) Data mining service: It is a central, transient Grid Service that is created by the data mining service factory. For different Data Mining Algorithm, different Data Mining Services are required. It is composed of instance service implementation class, resources class, and resources home class. The properties of Resources class can express some attribute, or example: supported algorithm, reusable software and hardware, status, error and result, support threshold, confidence threshold. It is classified as centralized and parallel-executed Data Mining Services.

4) Data mining workflow engine service: It congregates a series of activity related to data mining as a job. The user can formulate his tasks as a business process in an appropriate, formal language[4]. By extending Web service workflow engine, the service first analyses XML-based task description language, secondly confirm the type and constraints of user desired service, thirdly query grid information service to obtain necessary information, and select service for user's task, then create the instance of service, finally execute the step of task in sequence or in parallel, and monitor the status of task. When task is finished, the result will be return to grid portal.

5) Data mining presentation service: It is a transient service that receives a machine-readable form of a data mining model as input and translates, renders or processes the model into different output formats. Such formats may include user-friendly representations of decision-trees, charts, association rules or interactive systems a user can use afterwards.

6) Data mining preprocessing service: It is applied before data mining service. Its main functions include data cleaning, integration, handling missing data and statistical noise, aggregations, subset selection and many more. For different data mining algorithm, different preprocessing service is required. The service is able to operate on top of multiple different data sources; this can be executed on data integration service.

7) Data integration service: Business data is partitioned and deployed on every Databases of server, these data may be heterogeneous. By data integration service, a logically unified virtual data table is built. Data integration service shield from the distribution and difference of joined data source. We build a virtual data source to implement data integration service by extending OGSA-DAI service.

2.3 Grid portal layer

Grid portal provide an interface of grid system for user. By grid portal, user can submit a new data-mining task, browse the status of task and visible presentation of data mining result and manage user. Under OGSA framework, combining web service workflow with grid task description, by workflow editor, end user first input desired data mining service, data mining presentation service, parameter, requirement about resource, desired dataset, QOS, data mining pattern. Then workflow editor transform these input to XML-based web service workflow and submit it to data mining workflow engine service, which will act as grid system resources agent to carry out the schedule of service instance and monitor of task execution, then return result of task to user when task is completed. Following design thinkings are used in developing data mining grid portal.

1) Inherit existing excellent development tools to avoid repeating work. We select extending Grid Sphere to develop data mining portal.

2) Develop a "white-box" framework: Framework users override base classes and "hook" methods, which require users familiar with core framework interfaces. Core framework interfaces are based on community standard API.

3) Make use of design patterns: Patterns provide solutions to commonly recurring software design problems, and provide common language that makes code easier to read and understand.

3 The implementation pattern of data mining service

(1) Centralized implementation pattern

Centralized data mining service is executed on single node(traditional data mining application enabled grid). Centralized data mining algorithm is as follows: according to workflow, workflow engine service perform some data preprocessing first, then perform data mining service in centralized database, finally perform data mining Presentation Service and return the result of data mining to user. Data mining service and data preprocessing service is operated on grid data service that virtualizes the physical location and layout of the dataset. When performing every service, workflow engine first browses the registry for available services. The registry answers with endpoint reference of service to workflow engine, then workflow engine service will send it to target service factory, and the target service factory will create corresponding service instance(data preprocessing or data mining service). Preprocessing service instance perform data cleaning, integration, handling missing data and statistical noise, aggregations, subset selection and many more, and write the result back to data resources. Because data preprocessing may be long running, the service usage is asynchronous and workflow engine service is being notified or in other contexts iteratively queries the properties for the instance's status.

After data preprocessing service have successfully accomplished, data mining factory service create data mining service instance again. Every grid service includes resources property; we can access property of grid resources by get Property, setProperty and Query Property interfaces. Data preprocessing service and data mining service use this resources property to provide corresponding attribute (e.g. supported algorithm, available hardware and software, perform status, error, result, support threshold, confidence threshold).

(2) Paralleled implementation pattern

In this pattern, two or more nodes simultaneously participate in the data-mining task. Here we take association rules mining (Apriori algorithm) for example to become parallel. By interpreting workflow, Data mining workflow engine service will know that this data mining service is parallel, so it use MDS service to query available calculating nodes on grid at present, according to required QOS, invoke GRAM service to allocate resources, invoke data mining factory service to create certain amount of data mining instance service, invoke data preprocessing factory service to create certain amount of data preprocessing instance service. Then, the preprocessing service on every partition is performed in a parallel way. After preprocessing service is completed, the data mining service on every partition is performed in a

parallel way [5].

The complexity of parallel data mining algorithm is the result of data mining software; this result is satisfied with data mining parameters. The result of merge may be not exactness: It may happen that although the association rules is satisfied with minimum support threshold and minimum confidence threshold in entire database, but is not satisfied in given server. Under this condition, the rule will not appear on this server. Finally, after confidence threshold and support threshold of this association rules being merged, the result will be inaccurate. Take association rules mining algorithm for example, In order to acquire accurate results, you may express supported count (like numerator, supporting record count of association rules) and total count (like denominator) as property of data mining service resources. In every server only compute numerator and denominator. After combination, then sum up numerator, denominator separately obtained by every instance, the data mining result in grid server is read into memory and combined together. At last the result is provided to user by grid portal. In processing job, monitoring service monitors the run-time status of every grid service instance and user's task processing.

4 The implementation pattern of data integration

We are structuring our discussion along with four different possibilities of data distribution: Single data source、federated data sources (horizontal partitioning)、federated data sources(vertical partitioning)、federated data sources (heterogeneous data sources schemes). For the case of single data source, the data is concentrated in one physical data source. We will expose the data source as data service extended from OGSA-DAI. The data of horizontal partition federation data source is distributed on different nodes physically by row, every node has the same table structure, storing different record. In order to obtain a single global data source description, it is needed to unite these tables distributed on different nodes, and ensure the transparency of partition. The data of vertical partitioning federation data source is distributed on different nodes physically by column, every node has different table structure, but there is same id(primary key)on the different partition of the same table. In order to obtain a single global table, it is needed to join these tables distributed on different nodes according to id. For heterogenous federation data source, their database schema may be different. This class of problems arises when attributes with the same semantics differ in name, description, application or data type, e.g. let D1 be an relational database and D2, D3 be XML databases.

In order to avoid building a new proprietary solution and reimplementing well-solved aspects of Grid Data services, we have decided to extend the free available OGSA-DAI Grid Data Service (GDS) reference implementation to provide a virtual data source (VDS). Offering the same metadata as a normal Grid data source, it can be used and integrated in existing applications quite easily to hide the distribution and heterogeneity of the participating data sources by reformulating requests against the logical schema of the VDS.

Our goal is to provide a mapping mechanism from physical pattern to logical pattern, permitting the transformation of the request of logic resources to physical module and transforming the result back. For the three federation patterns, we design data integration as middleware service, linking participated data source, integrating them as a logically virtual data source, sending query to them and receive result from them in flexible way. We use an XML-Script for describing data integration service and explain how mapping original data source to target virtual table. For horizontal partitioning federation data source, target table name, primary key, original field and target field of union operation must be appointed in configuration files；For vertical partitioning federation data source, target table name, primary key, original field and target field of join operation must be appointed in configuration files; for heterogenous federation data source, corresponding combination of union and join operation must be specified in configuration files.

A client submits perform document, which include various activity to data integration service. Then data integration service first checks the query, e.g. if the used attributes are available and if it is a valid statement. Next, it appoints the relevant data sources with the help of the mapping schema in configuration file, and develops the query execution plan to obtain the requested data. For example, a SELECT-statement is executed on the relational database and two different XQUERY/XPATH-statements are performed on the XML databases. The results of the XML databases are joined via the information provided in the join-element of the mapping schema, and the two result-sets are merged together as specified by the union-element in the mapping schema.

5 Conclusions And Future Work

The integration of data mining and grid is discussed in this paper. The five layers architecture of GT4 and OGSA-DAI-based data mining is put forwarded. Design ideas of data mining portal is discussed, then centralized and parallel implementation pattern of data mining service is set forth. Finally the implementation pattern of data integration is brought forward. This discussion about data mining service here takes association rule as examples. This mining technology is appropriate for coarse granularity parallel. For other data mining technology, for example, cluster, classification and regression, appropriate parallel arithmetic must be researched.

6. REFERENCES

[1] T.C.Li, T.Bollinger, et al, Grid-based Data Mining in Real-life Business Scenario Germany, 2004.

[2] T. Harmer, J. McCabe, GT3.2 to GT4 Migration: A First HOW TO Release 0.1 The Structural Issues. Jan 7, 2005.

[3] B. Peter, A.M. Tjoa, et al, Grid Miner: An Infrastructure for Data Mining on Computatutional Grids, 2004.

[4] M. Ong, X. Ren, et al, Workflow Advisor on The Grid, United Kingdom.

[5] A. Depoutovitch, A. Wainstein, Building Grid-Enabled Data-Mining Applications Tackling big Data-Mining jobs with parallelization, 2005.

Realization Mechanism of the Business Collaborative Knowledge Discovery System Based on Distributed Data Environment*

Jianming Lin, Chunhua Ju
Zhejiang Gongshang University,Hangzhou, Zhejiang, China
Email: linjianming@czt.zj.cn,juchunhua@hotmail.com
Wenxiu He
Zhejiang University of Technology,Hangzhou, Zhejiang, China
Email: wenxiuhe@zjut.edu.cn

ABSTRACT

Based on the research of distributed data mining, some work was done on architecture design and realization method of business collaborative knowledge discovery system. To obtain valuable knowledge from business distributed data resources and to realize the integration of knowledge discovery and knowledge management, this paper presents an integrative architecture of distributed knowledge discovery and knowledge management, designs the realization flow of distributed knowledge discovery based on the new architecture. Through a practical example, it analyzes and illustrates the designing methods and realization mechanism of the business collaborative knowledge discovery system.

Keywords: Distributed, Knowledge Discovery, Collaborative, Realization Mechanism.

1. BACKGROUND

With the development of information technology and the acceleration of globalization, large-scale business enterprises, especially chained commercial enterprises, are developing toward globalization management depending on the Web and information technology, and commodity trading is changing into information trading. A great deal of branches, distribution centers and headquarters form a shared and distributed business data environment through Internet. These distributed business data increase at a rate of hundreds megabytes every day, thus a large-scale business data gold mine comes into being. It is very significant to mine the valuable knowledge from the mass distributed and quickly changed business data for analyzing customer and market, improving market decision-making and enhancing enterprise's kernel competitive ability.

Recently, there are fruitful researches on distributed data mining. In theory, paper [2,3,4] present some concepts and methods, including meta-learning, coactive-learning, knowledge exploration for data mining of homogeneous nodes, paper [1] presents the concept of CDM (Collective Data Mining); in architecture model, paper [5] presents an abstract model of distributed data mining system, MOIRAE (Management of Operational Interconnected Robots using Agent Engines), paper [6] presents model Sybil which constructs distributed data mining system under heterogeneous environment; in distributed algorithm, paper [7,8] studies the creating algorithm of distributed decision tree and the mining algorithm of distributed association rules respectively; in application, paper [2,5,9] designs and implements some typical distributed data mining system, such as JAM system, DIDAMISYS system and PADMA system.

However, most researches done emphasize particularly on the model architecture and algorithm implementation of distributed data mining and the research mainly focus on how to achieve the assignment allocation and coordination between global mining and local mining. There are still few complete theories and instances as to how to hierarchically organize and manage the knowledge and result mined from distributed data mining, how to introduce the architecture of knowledge extraction, knowledge integration, knowledge evaluation, knowledge searching and knowledge application, how to implement KD (knowledge discovery) with Internet collaboration under distributed data environment. Taking the

* This work was supported by the grand foundation of National Social Science Fund under Grant No 05BTJ019, foundation of National Doctor Fund under Grant NO 20050353003.

distributed database of commercial enterprises as example, this paper makes some efforts on architecture design and implementation system of business collaborative KD.

2. DESIGN OF ARCHITECTURE AND FUNCTION

As Fig. 1 shows, the architecture of the distributed collaborative KD system consists of KD platform, knowledge management platform, KD Agent, knowledge base and the interactive human-machine interface. The primary components and their function is introduced as follows:

Knowledge representation and interactive human-machine interface: it is a visible interface the user interacts with the system. The decision maker puts forward knowledge request within its authorization and do the interactive KD and knowledge visiting. The results of KD and the knowledge searched from knowledge base are presented to the user friendly.

KD Platform: it is the assignment manager and coordinator of the entire collaborative KD system. Of all the components of the platform, Assignment Interpreter is responsible for interpreting, decomposing, allocating assignments and processing knowledge request. Agent Creator adopts present model and algorithm to create KD Agent according to assignment. Assignment Scheduler and Communication Coordinator is used for the allocation of Agents, the data exchange, communication and coordination

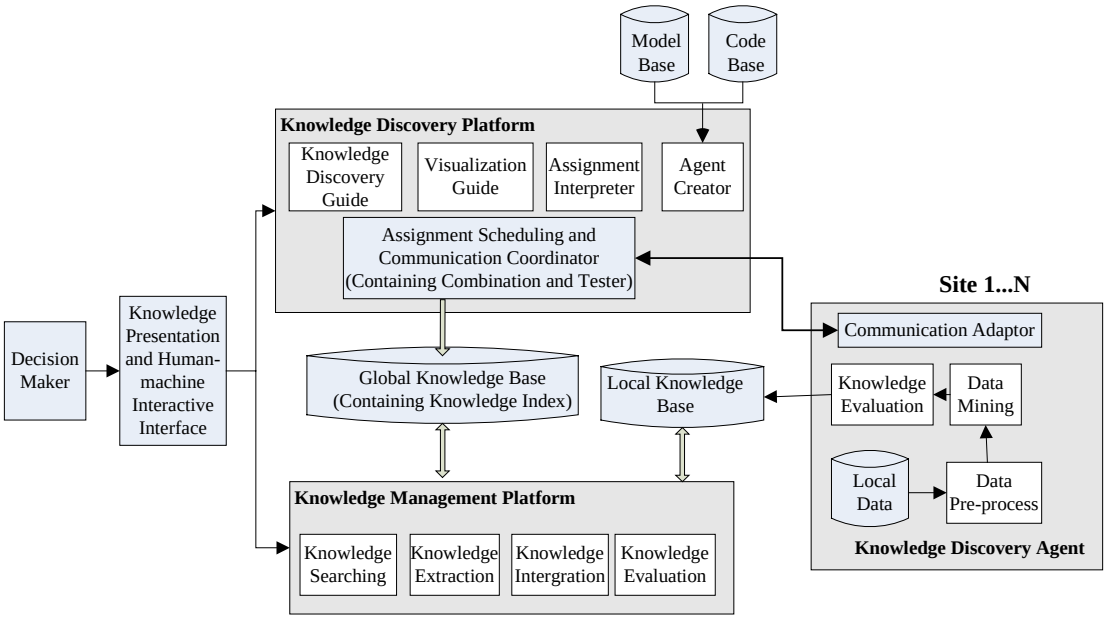


Fig. 1 Architecture of Distributed Collaborative Knowledge Discovery

between the Agents during working with each other. Combination and Tester combine and test knowledge and combine the local rules created at different sites into general class.

KD Agent: it implements KD assignments scheduled by Assignment Scheduler and Communication Coordinator. At the course of KD, data preprocess is responsible for data scrubbing, data integration, data selection and data transformation. Data mining recognizes data pattern, i.e. discovers some knowledge, such as association rule, classification rule, data aggregation, serial pattern, similar pattern and chaos patter. Knowledge Evaluation evaluates and interprets the results mined from databases.

Knowledge Base: it includes global base and local base. It stores the interested knowledge extracted from distributed data source, which can be the reference of decision-making. In order to be understood and searched easily, the knowledge should be stored in the form of clear hierarchy, structure and causality.

Knowledge Management Platform: it maintains, organizes and manages global knowledge base and local knowledge base, calling search engine to search knowledge, synchronizing and updating knowledge from local base to

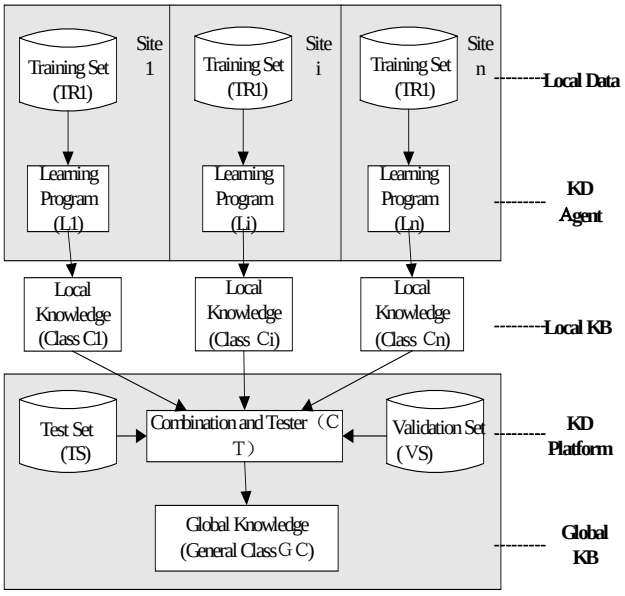


Fig. 2 Flow of Distributed Collaborative KD

global base by extraction and integration, and so on.

3. FLOW DESIGN OF COLLABORATIVE KD

Fig. 2 shows the knowledge extraction process of distributed and collaborative KD. Site 1 to n take local data as training set, run the learning program to train and learn simultaneously, extract basic local rules. The Combination and Tester creates General Class (GC). The whole process is described as follows:

Step 1: Agent Creator in KD Platform designs and creates corresponding Agent (Learning Program) according to the target and requirement of KD, each site prepares its own training set TR1…TRn by data preprocess.

Step 2: Training set is trained by learning program Li at each Node i (i = 1…n), then we get Class Ci. Step 3: KD Platform extracts TS (Test Set) and VS (Validation Set) through training set at each site;

Step 3: CT combines Class C1…C2 (which is created at different site), then it gets GC, and GC is tested by TS and validated by VS.

4. DEMO —— COLLABORATIVE KD OF CUSTOMER CLASSIFICATION BASED ON SPLINT ALGORITHM

4.1 Demo-data and Problem Description

Table 1 shows the customer demo-data of the research stored at Site A and B. The information on two attributes of Age and Income is stored at Site A and the information on

Table 1. Customer's Demo-data

RID	Age	Income	Buys_ Computer	RID	Is_ Student	Credit_ rating	Buys_ Computer
1	48	high	no	1	no	excellent	yes
2	53	high	yes	2	yes	fair	yes
3	35	medium	yes	3	yes	bad	yes
4	25	low	no	4	no	fair	no
5	22	low	yes	5	yes	excellent	yes
6	43	low	yes	6	no	excellent	no
7	32	medium	yes	7	no	bad	no
8	36	low	no	8	no	fair	yes
9	37	high	yes	9	yes	fair	no
10	51	medium	no	10	yes	bad	yes
		Site A				Site B	

Table 2. Attribute list of Age at Site A

Age	Buys_computer	RID
53	yes	2
51	no	10
48	no	1
43	yes	6
37	yes	9
36	no	8
35	yes	3
32	yes	7
25	no	4
22	yes	5

Table 3. Count Matrix of Credit_rating at Site B

Credit_rating	Buys_computer	
	yes	no
excellent	2	1
fair	2	2
bad	2	1

whether the customer is a student and customer's Credit_rating is stored at Site B. We want to get the classification rule between the user's personal information and the status of buying computer (Buys_computer). In essence, it is a heterogeneous distributed collaborative KD problem.

4.2 Creating Training Set

Splint algorithm sorts the numerical continuous attributes first, and then stores the sorting results by attribute list or histogram. Table 2 shows the attribute list of attribute Age at Site A.

To discrete classification attributes, the attribute list or histogram can be denoted as count matrix to show the distribution of each attribute intuitively. Table 3 shows the count matrix of attribute Credit_rating at site B.

4.3 Learning Process of Agent

After the training sets at each site are prepared, KD Agent begins to run learning program: calculating the best splitting scheme. In Splint algorithm, how to split certain site best is measured by Gini index. This kind of measure method can be described as follows: if set T contains m records of n classifications, then its Gini index is

Gini(T)=1- $\sum_{j=1}^{n} p_j^2$ (p_j is the frequency of class j appeared) Eq.(1)

If set T is divided into two parts, T1 and T2, contains m1 and m2 records respectively, then the Gini index of the partition is:

$$gini_{partition}(T) = \frac{m1}{m} gini(T1) + \frac{m2}{m} gini(T2) \quad \text{Eq.(2)}$$

The Gini index providing the smallest partition is selected as the best partition of set T.

Take the attribute list of Age as an example (showed in table 2), there are 10 records and perhaps 9 partitioning nodes at most. For the first partitioning node, meaning to partition the first record with the other 9 records, C_{below} is (yes=1,no=0), C_{above} is (yes=5,no=4) (C_{below} and C_{above} is the class distribution histogram of attribute Buys_computer). Therefore, The Gini index of position1 is calculated as:

gini(position1)=

$$\frac{9}{10}\times\left[1-\left(\frac{5}{9}\right)^2-\left(\frac{4}{9}\right)^2\right]+\frac{1}{10}\times\left[1-\left(\frac{1}{1}\right)^2-\left(\frac{0}{1}\right)^2\right]=\frac{40}{90}=0.444$$

Also, we can calculate the Gini indexes from the second partitioning node to the nineth node, 0.475, 0.419, 0.467, 0.48, 0.45, 0.476, 0.475, 0.444(the best partitioning is at the third partitioning node and the Gini index is 0.419)。

The histogram of attribute Credit_rating only has two partitioning nodes and the Gini indexes are the same, 0.476.

The histogram of attribute Income of Site A also has two partitioning node and the Gini indexes are 0.476 and 0.467 respectively.

The histogram of attribute Is_student of Site B only has one partitioning node and Gini index is 0.4.

4.4 Synthesizing Knowledge

The Agents at each site send the calculated Gini indexes of each attribute and the optimal partitioning result to KD Platform, which will select the smallest Gini index as the global partitioning attribute. In this case, attribute Is_student at Site B is selected as root node during the first partitioning and the global classification knowledge (decision tree) after first partition is as Fig. 3 shows.

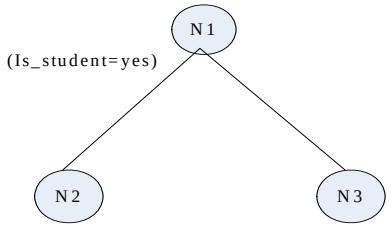


Fig. 3 Global Classification Knowledge after First Partition

4.5 Description Of Collaborative KD Algorithm

The above is just the full process of one partition. Next partition begins after first partition. At this moment, each site rescans attribute list according to RID value of the Hash table, extracts the records belonging to the same node to create new attribute list and histogram and calculates its Gini index and decides the next partitioning scheme. The whole classification KD process is described as follows:

MakeDecisionTree（Di，Aij，Li，k） // Di is the data set of site I, Aij is the j_{th} classification attribute, Li is the attribute number of Site i, k is the number of sites.

create training set Si according to Di (attribute list or histogram);
create node queue and put it in M;
while queue is not null{
take out the first node N from queue;
if node N meets end condition then tag it as leaf node, continue;
for （i=1，i<=k，i++）{
for（j=1，j<=Li，j++）{
calculate Gini index of Aij;
decide the optimal partitioning scheme of Aij; }
decide the smallest Gini index and local optimal partitioning scheme of site i according to Gini index and the optimal partitioning scheme of each Aij(J=1,…,Ni);

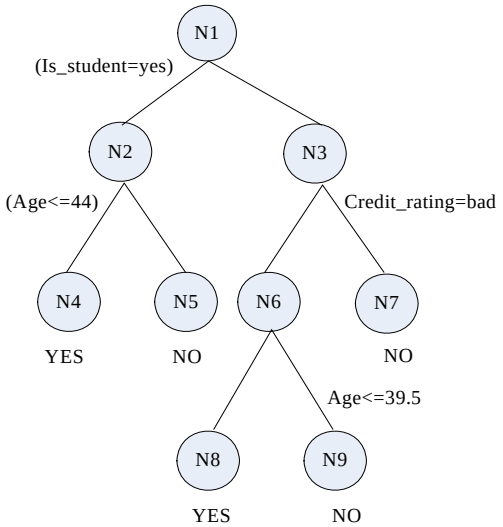


Fig. 4 Global Classification Knowledge

send Gini index and the optimal partitioning scheme to KD Platform; }

KD Platform selects the optimal partitioning point F*, N creates branch node N1 and N2, then put them in queue;

Partition the attribute list of the partitioning point and create the Hash table containing the records according to the RID;

Each site partitions other attribute list according to Hash table and creates attribute histogram;}

The complete global decision tree is as Fig. 4 shows.

5. CONCLUSIONS

At present, thousands of databases are used in government, business, bank and insurance and form many distributed data gold mine. However, the problem of "great deal of data, short of knowledge" still exists widely. Based on the related research of distributed data mining and the study of the collaborative KD and knowledge management under distributed data environment, this paper presents an integrative architecture of distributed KD and knowledge management, designs the implementation flow of distributed KD based on the new architecture and analyzes the designing methods and implementation mechanism of the business collaborative KD system through an practical example. By the way, this paper focuses on the architecture and flow design of distributed knowledge discovery and we will make further research on related subjects such as collaborative mechanism, knowledge extraction and knowledge integration.

6. REFERENCES

[1] H. Kargupta, B. Park, et al, "Collective data mining: a new perspective toward distributed data mining", In: Kargupta H, Chan P. Advances in Distributed and Parallel Knowledge Discovery. Menlo Park, CA/Cambridge, MA: AAAI/MIT Press, 2000.

[2] A. Prodromidis, P. Chan, et al, "Meta-learning in distributed data mining systems: issues and approaches", In: Kargupta H, Chan P. Advances in Distributed and Parallel Knowledge Discovery. Menlo Park, CA/Cambridge, MA: AAAI/MIT Press, 2000.

[3] D.L. Grecu, L.A. Becker, "Coactive learning for distributeddata mining", In: Piatetsky-Shapiro G, Agrawal R, Stolorz P. Proceedings of the 4th Intl-Conf on Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press, 1998.

[4] Y. Guo, J. Sutiwaraphun, "Probing knowledge in distributed data mining", In: Zhong N, Zhou L. Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge discovery and Data Ming. Berlin: Springer, 1999.

[5] J.M. Pena, E. Menasalvas, "Towards flexibility in a distributed data mining framework",In: Proceedings of the ACM SIGMOD Workshop on Research Issues in Parallel Knowledge Discovery. Santa Barbara, CA, 2001.

[6] R. King, M. Novak, et al, "The sybil database integration and evolution environment: an overview", In: Proceedings Of the 1st East-European Symposium on Advances in Databases and Information Systems. St. Petersburg, Russia, 1997.

[7] J. Shafer, R. Agrawal, et al, "Sprint: A scalable parallel classifier for data mining", In: Proceedings of the 22nd intl. Conf. On Very Large Databases. Mumbai, India, 1996.

[8] R. Agrawal, J. Shafer , "Parallel mining of association rules: design, implementation and experience", Research Report RJ 10004, IBM Almaden Research Center, San Jose, CA, 1996.

[9] H. Kargupta, I. Hamzaoglu, et al, "Scalable, Distributed data mining-an agent based architecture", In: Heckerman D, Mannila H, Pregibon D, Uthurusamy R. Proceedings of the 3rd Intl. Conf. On Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press, 1997.

Application of Image Data Mining Techniques In Fire Detection

Ting Li
School of Information Technology, Southern Yangtze University
No.1800, Lihu Dadao, Wuxi Jiangsu 214122, China
Email: mouselisa@sina.com

ABSTRACT

Image data mining techniques play an important role in fire detection. In the paper, the procedure of fire recognition and the significance of fire detection based on image data mining techniques is discussed. The method of image front-end acquisition, feature presentation, feature retrieval and image recognition is introduced. We describe the method of judge the fire through the way of feature retrieval, feature recognition and similar search on the fire image.

Keywords: fire detection, image data mining, model identifying.

1. INTRODUCTION

Fire detector is an important part of fire detection and alarm system, which is widely used in residential building, industrial plants and platforms and public facilities areas. Current fire detectors methods are mainly dependant on its sensor and those based on the analysis of computer images [1,2,3,4,5]. Its type includes smoke fire detectors, heat detectors, flame detectors and so on. As sensor's limitation, it is passive to distinguish fire accident. It cannot actively distinguish fire accident real situation and fault situation. Based on the image data mining, we use the methods of image search and pattern reorganization to analysis the scene and obtain the valid feature data of fire. This method can judge whether the fire exists intelligently.

2. TECHNIQUES IN THE IMAGE DATA MINING

2.1 Feature presentation

The most concern is image visible characteristics acquisition and expression. Different expression methods of image characteristics take various different points of view to portray this characteristic's some properties. The most concern is standard visible characteristics, which include color characteristic, vein characteristic, shape characteristic and so on.

Color characteristic is widely used visible characteristic in image search. Color characteristic mainly includes color histogram, Color Square, color volume, color aggregation vector, color correlation diagram and other related color characteristic expression method.

Vein characteristic does not rely on color or brightness. Vein characteristic mainly include Tamura vein characteristic, autoregressive vein characteristic model, directivity characteristic, wavelet transform, co-occurrence matrix and other forms.

Shape characteristic is another important characteristic in image expression and image search. General speaking, shape characteristic has two expression methods. One is profile characteristic. Another is area characteristic. The classical method of the two-shape characteristics is Fourier descriptor and shape non-related square.

2.2 Feature acquisition

Image visible characteristic can be regarded as image's characteristic semantic. Image characteristic semantic acquisition and processing is easy. It can be done through digital image handling technology. It mainly includes color, vein, shape, chroma, brightness, saturation and other related information.

2.3 Object reorganization

In image semantic, image object identification can be summarized as image object semantic acquisition. Its involved key technologies include: image segmentation, object model expression and object identification. The most important contribution part in image object semantic acquisition concentrates on image meaningful part. After separate meaning area in the image, area's object segmentation and identification will be done.

2.4 Image similar search

Text-based searches adopt text accurate matching. But content-based image searches calculate and query comparability matching between example image and candidate image. Therefore, it has great impact on searches effect concerning define one proper visible characteristic comparability measurement method. Most image visible characteristics can be expressed in vector form. Typical comparability methods are all vector space models. That is, treat visible characteristic as the point in the vector space. Calculate the adjacencies degree to measure comparability in the image characteristics. In early 1990's, content-based image retrieval, CBIR, technology has been proposed. CBIR adopts image's low level visible characteristics, such as color, texture, shape, or wavelet coefficient to do comparability searches towards image identification. Famous CBIR model and system includes QBIC, Virage, MARS and so on.

3. APPLICATION OF IMAGE DATA MINING IN THE FIRE DETECTION

Fire accident is a burning process. There will be a lot of black smoke, carbon granule, fly ash and other suspended particles in the space when fire accident happens. The flame happens and the temperature increases. Based on the CCD equipment, the space images can be put into storage equipment to form original image volume. Computer does standardization work towards original image volume. That is, normalize all parameters, such as image brightness, saturation and delete any images difference caused by image acquisition equipment error. Then acquire image visible characteristics (color, texture), identify image object, find

out meaning area, finally match identified interesting image with preset fire characteristics through image comparability searches. Therefore confirm whether there is fire characteristic existing or not. The system flow chart is shown in Figure 1.

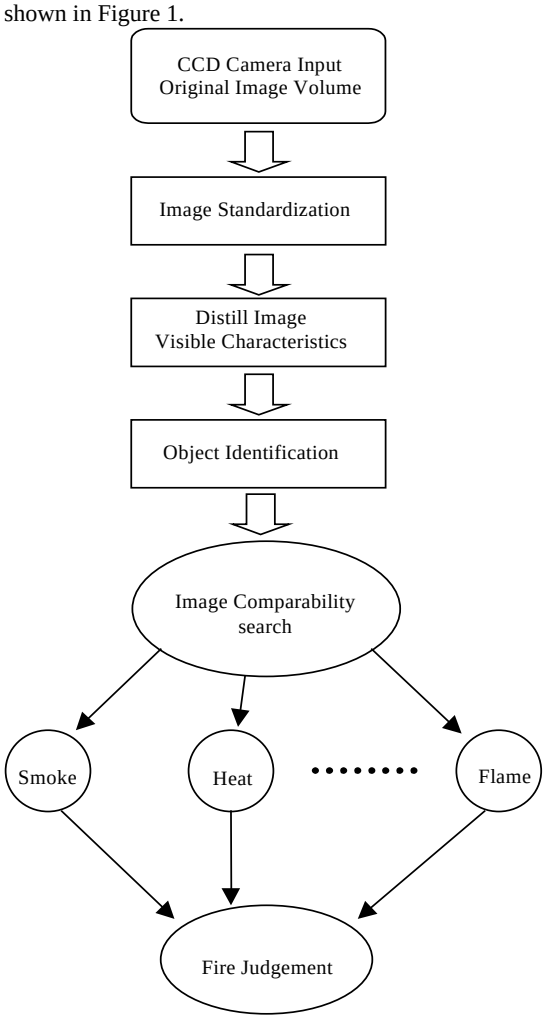


Fig. 1. system flow chart

3.1 Acquire color characteristic

Standardization-made digital image adopts RGB color space. The weights represent red, green and blue respectively. This structure does not comply with person subjective judgment towards color comparability. Normally histogram adopts HSV space as color space. The weights represent Hue, Saturation and value respectively. Transformation formula from RGB space to HSV space is shown as:

v=max(r, g, b)
s=(v-min(r, g, b))/v
h=5+b' if r=max(r, g, b) and g=min(r, g, b)
h=1-g' if r=max(r, g, b) and g!=min(r, g, b)
h=1+r' if g=max(r, g, b) and b=min(r, g, b)
h=3-b' if g=max(r, g, b) and b!=min(r, g, b)
h=3+g' if b=max(r, g, b) and r=min(r, g, b)
h=5-r' otherwise
r'=(v-r)/(v-min(r, g, b))
g'=(v-g)/(v-min(r, g, b))
b'=(v-b)/(v-min(r, g, b))

where r, g, $b \in [0 \ldots 1]$, $h \in [0 \ldots 6]$, and s, $v \in [0 \ldots 1]$.

3.2 Distill texture characteristic

Decompose standardized image signal into a series basic function $\Psi mn(x)$. These basic functions are got from generating function transfiguration as follows:

$$\psi mn(x) = 2^{\frac{m}{2}} \Psi\left(2^{-m} x^{-n}\right) \quad (1)$$

Where m and n are integer. Then $f(x)$ can be expressed as:

$$f(x) = \sum_{m,n} c_{mn} \Psi_{mm}(x) \quad (2)$$

According to frequency characteristics, they are called as LL, LH, HL and HH respectively. Decompose pyramid-structured wavelet transform (PWT) and tree-structured wavelet transform (TWT), PWT recursively decomposes LL wave band. But towards those texture characteristics mainly included into middle frequency band range, it is not enough to just decompose low frequency LL wave band. Besides recursively decompose LL wave band, TWT also can decompose other LH, HL and HH wave bands. It is treated as one supplement for PWT. After decomposing various wave bands, distill image texture characteristic.

Based on color histogram, acquire smoke, heat and flame color characteristics from standardized image. Distill smoke, heat and flame texture characteristics from wavelet transform. Then, use object identification method to distill meaning area. Finally based on histogram intersect, 2nd distance or Mars distance methods, conduct comparability matching between these meaning data and preset fire characteristic data. These processes have many mature technologies and arithmetic. The fire characteristic data can be acquired in fire testing room. According to specific arithmetic, establish smoke, heat and flame visible characteristic models.

4. CONCLUSIONS

In the paper, we have proposed a supposition to the intelligent fire detection method based on the image data mining. The image data mining theory and the procedure of processing the collected images is introduced. In the future, the research and the development on the image data mining will expand in the scientist areas.

5. REFERENCES

[1] F.N. Yuan, G.X. Liao, et al, "Feature extraction for computer vision based on fire detection", Journal of university of science and technology of china, vol.36, No1, Jan 2006, pp.39-43.

[2] H.Y. Yuan, B.H. Liu, et al, "Image based fire intelligence detection and special location", Fire protection science and technology, vol. 2, 1998, pp. 224.

[3] X. Cheng, D.C. Wang, et al, "Image Type Fire Flame Detecting Principle", Fire Safety Science, Vol.14, No.4, Oct 2005,pp. 239-245.

[4] H.B. Jin, "Fire Detection Technology Based on Digital Image Processing", Fire Science and Technology, Vol.3, May 2002, pp. 46-47.

[5] H.C. Ran, "The Fire Picture Detection Method of the Fuzzy Classified Recognition", Fuzzy System and Mathematics, Vol.15, No.3, Sept 2001, pp. 102-104.

Query Model of Heterogeneous Database of bridge Collaborative design system

Ming Chen
School of Engineering and Architecture, Shanghai Institute of Technology, Shanghai, China, 200235
Email:chenmchen@21cn.com

Abstract

This paper describes a prototype implementation of query model for bridge collaborative design system. The model incorporated a heterogeneous database as the backend to store analysis results, and the Internet were utilized as communication channel to access the analysis results. Three key issues regarding the query model were discussed, including system architecture, global agent, and local agent. The query model gives great flexibility and extendibility to the data management in bridge collaborative design system and can provide additional features to enhance the applicability of bridge design software.

Keywords: Heterogeneous Database, Bridge design, Collaborative Design.

1 Introduction

A typical project of bridge design involves a wide range of disparate professionals — architects, structure engineers, building services engineers, etc. Along with the current growth of the Internet and collaborative design approach in bridge design, it emerges as an important research issue to access distributed heterogeneous databases. To retrieve information from numerous data sources, the system is needed to integrate various resources and process queries in a distributed manner.

The problem of transaction management in heterogeneous databases consists of developing a software module, on top of local database management systems, that allows users to execute transactions that span heterogeneous databases without jeopardizing the consistency of the data. The difficulty arises due to the following two characteristics of the heterogeneous database system (MDBS) environments: Heterogeneity and Autonomy.

In this paper, we develop a framework for designing fault-tolerant transaction search algorithms for autonomous MDBS environments. This paper is organized as follows. Section 2 introduces the technology and system architecture of integration of heterogeneous databases. In Section3 and 4, we describe the methods of agent design. The experimental results are shown in Section 5. Concluding remarks are made in Section 6.

2 System architecture

Most of approach in integration of heterogeneous database systems area to date has based on the technology of Object Oriented, CORBA, XML and so on. Most researchers generally put more focus on product model, consistency, and system architecture, while the implementation of system model, e.g. the agent design, integration management of heterogeneous data, has not received enough attention.

System architecture is the software organization and construction of heterogeneous database, which decides which system characteristics to use, which could provide the greatest convenience and flexibility. We summarize three kinds of system mode for heterogeneous database integration: integrated mode, distributed mode and integrated-distributed mode.

In integrated mode the distributed users register with the server and operate the system remotely. Recently, extensive research and development works have been carried out to develop prototype system and methodologies for heterogeneous database integration based on integrated mode [1, 2, 3, 4, 5]. The most obvious feature of distributed mode is its flexibility, but without a central server many model interpreters are required between different domain systems. This Integrated-distributed mode can be concluded as the open system and has some features i.e., heterogeneous platforms, system flexibility, and system stability. Our approach is based on this mode, using the latest software component technology and agent technology to design an open integration for heterogeneous database. Fig. 1 presents the general prototype of Heterogeneous Database of bridge collaborative design system Based on Agents.

Fig. 1 illustrates the overall architecture of Heterogeneous Database integration. The system is built on a browse – server architecture and the users access the heterogeneous database on the Internet through its visual query interface developed as a Java client Applet.

Application, DBMS and local database are part of existing system. Data agent composed of global agent and local agent is lies in the heart of the system. It is responsible for answering user queries and storing design data in a collaborative environment. Steps of the heterogeneous database search are as follows:

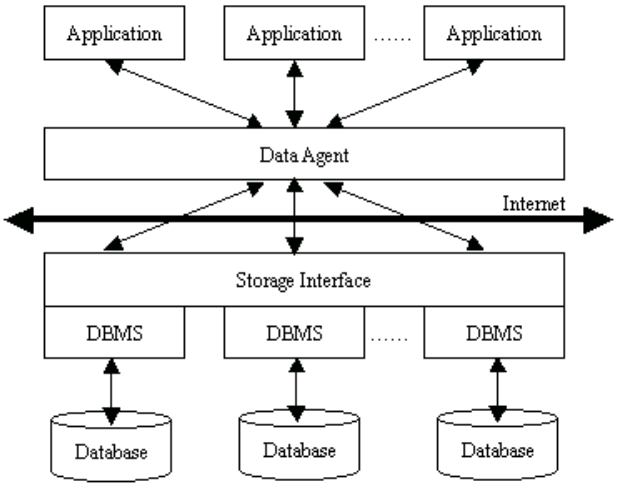


Fig. 1. Prototype of heterogeneous database integration

Step 1 Application send search command *Query* to integration system.

Step 2 Global agents are activated:

Step2.1 Search disassemble： The global agent

divides *Query* into query set C ($C = \{id, q_1, q_2, ..., q_n\}$) and result tree $Tree_q$ - id is search no generated by global agent automatically, which is unique during system. The q_i is Sub- query command, which includes Destination address ip_i .

Step2.2 Format transform：Global agent combines q_i with id , and then translates it into standard query command s_i (XML)

Step 3 transmit s_i to local agent through Internet

Step 4 Local agent is activated:

Step 4.1 local Agent retrieves q_i from s_i and transforms q_i into local format that can be identified by local DBMS, then send query command to local database.

Step 4.2 local Agent retrieves id from s_i and reserve id in pre-definition variables of local agents, then let the state of local agent be *busy*.

Step 5 According to request from local agent local DBMS query database and return search result to local agent.

Step 6 Local agent is activated:

Step 6.1 Get search result from local database。

Step 6.2 Retrieve id from pre-definition variables of local agent.

Step 6.3 Combine search result with id and translate it into standard data flow t_i .

Step 6.4 let the state of local agent be *free*.

Step 7 transmit t_i to global agent through Internet

Step 8 Global agents are activated:

Step 8.1 Transform t_i into return information flows r_i , and classify r_i based on id

Step 8.2 Retrieve $Tree_q$ of id ,

Step 8.3 Generate *result* according to r_i and $Tree_q$.

Step 8.3 Translate *result* into local format and return it to application.

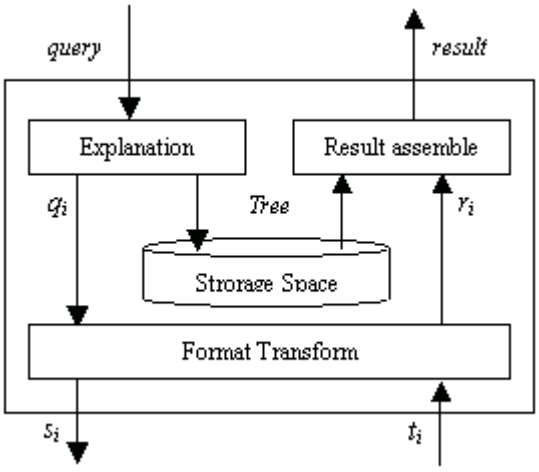


Fig. 2. Global agent

3 Global Agent

Global agent is responsible for managing request from application and returning result from database. Global agent (Fig. 2.) is composed of semantic explanation model for ESQL, format transform model, and inner storage space and result combination model.

3.1 semantic explanation model for ESQL

Semantic explanation model for ESQL is lies in the heart of the global agent, which influence agent's realization and working efficiency.

1) ESQL. ESQL is an extended SQL that is better suited for heterogeneous database query tasks than traditional SQL.

Definition 1(Operator of ESQL Set)： Let O be set of SQL operator $O = \{\cup, \cap, —, \times\}$, where The $\cup$ means union, the union of two SQL languages sql_1 and sql_2 is the set obtained by combining the result of each without repetition. The $\cap$ means Intersection, the intersection of two ESQL languages sql_1 and sql_2 is the result common to both.

— means subtraction, the subtract of two ESQL languages sql_1 and sql_2 defines a new set, get by removing from the result of sql_1 it has in common with the result of sql_2 .

× means Cartesian product, the Cartesian product of two ESQL languages sql_1 and sql_2 is defined to be the result of all points (a,b) where $a \in A$ and $b \in B$, where A is the result of sql_1 and B is the result of sql_2 .

The operator of ESQL set is different from the operator in Relational algebra. The ESQL operator is the operation among the result of ESQL language rather than the ESQL language.

Definition 2(Operator of ESQL Logical)： Let R be set of SQL relation $R = \{\&\&, \|, \neg\}$, where the $\&\&$ means AND, The logical AND operation compares results of sql_1 and sql_2 , and if they are both *true*, then the result is *true*; otherwise, the result is *false*. The $\|$ means OR, The logical OR operation compares results of sql_1 and sql_2 , and if either or both results are *true*, then the result is *true*; otherwise, the result is *false*. The $\neg$ means NOT, The logical NOT operation simply changes the value of a single result of sql_1 . If it is a *true*, the result is *false*; if it is a *false*, the result is *true*.

Definition 3(*Address* Clause)： The query allows the specification of target databases in the *Address* clause. The global agent then transforms the federated query into sub-queries of local databases.

Definition 4(*ESQL* Expression)： *ESQL* Expression is a serial *ESQL* language connected through Operator of *ESQL*.

2) Semantic explanation model. Semantic explanation model is built on J2EE system for performing all common tasks of explaining ESQL language. The output information of semantic explanation model includes $Tree_q$ and q_i , where $q_i = \{id, ip_i, KernelText_i\}$ is sub-queries set:

Semantic explanation model is built on J2EE system for performing all common tasks of explaining ESQL language. The output information of semantic explanation model includes $Tree_q$ and q_i, where $q_i = \{id, ip_i, KernelText_i\}$ is sub-queries set:

id: is query No., which is the same to every sub-query generated from a query. The ip_i is the IP address of the target database of sub-query i. The $KernelText_i$ is kernel information of q_i, which includes parameter of sub-query.

$Tree_q$: is query result tree, which generated through query, disassembling in Step2.1.

Semantic explanation model is realized through five functions: GetQuery, AllocateId, ExplainQuery, StoreTree and SendQuery.

3.2 formats transform

Format transform of global agent is a bi-directional module, which realizes by the two functions: TranslateQueryToXML: it transforms local query command into global query command. The result is a query flow of XML format and TranslateXMLtoResult.

4 Local Agent

Local agent is responsible for querying local database and returning result to global agent. Every local database has special local agent. Format transform of local agent is also a bi-directional module, which realizes by two functions: TranslateToLocalDBL and TranslatetoResult.

5 experiments

We have evaluated our heterogeneous database integration approach using several sets of real-world heterogeneous database used in bridge seismic collaborative design. Bridge seismic design involves three groups including bridge design engineers, seismic engineers and seismic design engineers.

Take the collaborative between seismic engineers and seismic design engineers as an example. As part of heterogeneous database seismic information is storied in local database of seismic engineers. Bridge seismic design can retrieve seismic information through the system interface.

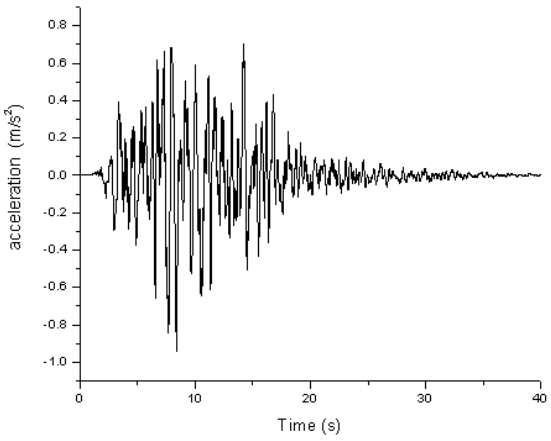


Fig. 3. Seismic information (time history)

Through system interface bridge seismic design engineers can get spectrum and time history information. Fig. 3. is a sample of seismic information.

6 Conclusion and Future work

Modern bridge collaborative design system is based on distributed systems with several independent databases. In such an environment, the integrated access to heterogeneous data stored in several more or less autonomous component databases remains an important problem. This paper proposes a model of heterogeneous database of bridge collaborative design system based on agents. The proposed framework used XML schema as the canonical command model for heterogeneous database integration. In this paper we presented algorithms for query, design approach of data agent, and illustrated them with examples.

Acknowledgements

This research reported herein is sponsored in part by Shanghai Nature Science Foundation through Grant Number 06ZR14079.

Reference

[1] E. Bertino, M. Negri, et al, “Integration of heterogonous database application through an object-oriented interface”. Information Systems, 1989, 4(5): 45-58.

[2] E. Bertino, “Integration of heterogonous data repositories International Workshop on Interoperability in Multi-database Systems”. Yoyo (Japan), Elsevier Press, 1991, 105-115.

[3] Z.D. Lu, B. Li, et al, “Study and Implementation of a Multi-database System Based on CORBA and XML” [J]. Journal of Computer Research and Development, 2002, 39(4): 443-449.

[4] S.B. Yoo, S.K. Cha, Integrity maintenance in a heterogeneous engineering database environment [J]. Data & Knowledge Engineering, 1997, 21(3): 347-363.

[5] Collins, S. Robert, et al, XML schema mappings for heterogeneous database access [J]. Information and Software Technology, 2002, 44(4): 251-257.