Journal of Computers

ISSN 1796-203X

Volume 7, Number 1, January 2012

Contents

Special Issue: Parallel Algorithms, Scheduling and Architectures

Guest Editors: Xianchao Zhang, Wenxin Liang, and Feng Xia

REGULAR PAPERS

Special Issue on Parallel Algorithms, Scheduling and Architectures

Guest Editorial

Since applications of computing systems permeated in every aspects of our daily life, the efficiency of execution of parallel programs on distributed systems has become a critical issue in the research field of high-performance computing systems. In recent years, more and more researchers have recognized the fact that parallel algorithms, scheduling and architectures play an important role in improving the efficiency of computing systems, and hence continuously present their valuable research results in this field.

In this special issue, we selected some excellent papers from the third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP 2010), which was held in Dalian, China, December 18-20, 2010. In addition, we invited and selected some representative research papers in the broad area of parallel algorithms, scheduling and architectures.

The paper titled "A Novel Differential Evolution with Uniform Design for Continuous Global Optimization" presents a uniform-differential evolution algorithm (UDE) which incorporates uniform design initialization method into differential evolution to accelerate its convergence speed and improve the stability.

The paper titled "Leveraging 1-hop Neighborhood Knowledge for Connected Dominating Set in Wireless Sensor Networks" proposes an algorithm leveraging 1-hop neighborhood knowledge for connected dominating set, aiming to get a small connected dominating set, meanwhile, to minimize the consumption of energy and time.

The paper titled “An Internet Traffic Identification Approach Based on GA and PSO-SVM" proposes an internet traffic identification approach which selects the best feature subset using Genetic Algorithm, and then calculate the correspondence weight of each feature selected by Particle Swarm Optimization (PSO). In addition, the traditional SVM algorithm is optimized by PSO algorithm.

The paper titled "Efficient and Scalable Thread-level Parallel Algorithms for Sorting Multisets on Multi-core Systems" proposes a cache-efficient, thread-level parallel and scalable algorithm for sorting Multisets on single multi-core computer and the heterogeneous cluster with multi-core computers.

The paper titled "A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment" presents a scheduling strategy on load balancing of VM resources based on genetic algorithm, brings in variation rate to describe the load variation of system virtual machines and introduces average load distance to measure the overall load balancing effect of the algorithm.

The paper titled "Task Scheduling Based On Thread Essence and Resource Limitations" discusses a task scheduling model takes into account the essence of the threads in simultaneously-running applications by granting higher priority to applications during their critical-serial phases and also considers the limited resources of the system by reducing the number of context swithches when there are more ready threads than cores.

The paper titled "A Game Theoretic Resource Allocation Model Based on Extended Second Price Sealed Auction in Grid Computing" proposes a model which focuses on the resource allocation problem in grids and cloud computing. This model introduces an analyst entity and designs analyst's prediction algorithm based on Hidden Markov Model and simulation results shows it performs better than traditional algorithms.

The paper titled “A Performance Model for Network-on-Chip Wormhole Routers” presents a generic analytical performance model of single-channel wormhole routes by using the M/D/l/B queuing theory.

The paper titled "Cocktail method for BitTorrent traffic identification in real time" proposes a cocktail approach consists of three sub-methods including application signature-based methods, message-based methods and pre-identification methods to identify BT traffic in real-time.

The paper titled “A Context-Aware Routing Protocol on Internet of Things Based on Sea Computing Model” proposes a new routing protocol CASCR (Context-Awareness in Sea Computing Routing Protocol) for Internet of Things, combining with the Sea Computing model for Internet of Things and the routing protocol for wireless sensor network (WSN) based on context-awareness, and describes the work flow, data structure and quantitative algorithm of the CASCR in details. Through theoretical and experimental analysis, the CASCR has higher energy efficiency and longer lifetime than the congeneric protocols.

The paper titled "An Energy-Aware Multi-Core Scheduler based on Generalized Tit-For-Tat Cooperative Game" proposes a game theoretic energy-aware scheduling algorithm for multi-core systems named GTFTES, which is designed to work in a resource-rich environment where resources always compete for tasks. Simulations results show that the proposed game can reduce the temperature difference between different groups of cores which effectively avoids the local hotspot of a processor.

The paper titled "A Failure Self-recovery Strategy with Balanced Energy Consumption for Wireless Ad Hoc Networks" discusses a solution of self-recovery strategy with balanced energy consumption in wireless ad hoc networks with the failure cluster which is out of work. Also, to keep the original performance of the whole WSNs, the energy consumption of the node is minimized.

The paper titled "An Attractive Force Model for Weighting Links in Query-Dependant Web Page Ranking" presents

doi:10.4304/jcp.7.1.1-2

a gravitation-like model for calculating the attractive force between papers. And then some features of web pages are taken into consideration while implementing an instance of the algorithm framework.

The paper titled "An Improved HITS Algorithm Based on Pagequery Similarity and Page Popularity" proposes a weighted HITS algorithm which differentiates the importance of links with the querypage similarities and the popularity of web pages. It can avoid the problem of topic drift and enhance the quality of web search effectively.

The paper titled “A Cooperative and Heuristic Community Detecting Algorithm” introduces the concept of community seed, vector and relation matrix. In terms of the relation similarity between free vertices and the existing communities, the authors put vertices into different groups. A minimum similarity threshold is proposed to filter which gives a method to find the vertices located at the overlapped area between different communities.

It has been a great pleasure to run this special issue, which reveals important research results in the field of parallel algorithms, scheduling and architectures. We would like to thank Prof. Prabhat Mahanti, Editor-in-Chief of Journal of Computers, and Dr. George J. Sun, Executive Editor of Academy Publisher, for giving us the opportunity to organize this special issue and for their great help in the organization of this issue. We thank all authors for their submissions and all reviewers for their diligent work in evaluating these submissions. We sincerely hope that you enjoy reading these distinguished papers.

Guest Editors:

Xianchao Zhang, School of Software, Dalian University of Technology, China
xczhang@dlut.edu.cn

Wenxin Liang, School of Software, Dalian University of Technology, China
liang@computer.org

Feng Xia, School of Software, Dalian University of Technology, China
f.xia@ieee.org

Xianchao Zhang is a Full Professor at Dalian University of Technology, China. He received his B.S degree in Applied Mathematics and M.S. degree in Computational Mathematics from National university of Defense Technology in 1994 and 1998, respectively. He received his Ph.D. in Computer Theory and Software from University of Science and Technology of China in 2000. He joined Dalian University of Technology in 2003 after 2 years of industrial working experience at international companies. He worked as Visiting Scholar at The Australian National University and The City University of Hongkong in 2005 and 2009, respectively. His research interests include Algorithms, Machine Learning, Data Mining and Information Retrieval.

Wenxin Liang an Associate Professor at School of Software, Dalian University of Technology, China. He received his B.E. and M.E. degrees from Xi'an Jiaotong University, China in 1998 and 2001, respectively. He received his Ph.D. degree in Computer Science from Tokyo Institute of Technology in 2006. He was a Postdoc Research Fellow, CREST of Japan Science and Technology Agency (JST) and a Guest Research Associate, GSIC of Tokyo Institute of Technology from Oct. 2006 to Mar. 2009. His main research interests include XML Data Processing and Management, XML Storage, Indexing, Labeling and Querying Techniques, XML Keyword Search, Web-based IR, Knowledge Discovery and Management, etc. He is a senior member of China Computer Federation (CCF), and a member of IEEE, ACM, ACM SIGMOD Japan Chapter and Database Society of Japan (DBSJ).

Feng Xia received the B.E. and Ph.D. degrees from Zhejiang University, China, in 2001 and 2006, respectively. He was a Research Fellow at Queensland University of Technology, Australia, from 2007 to 2008. Before joining Dalian University of Technology (DUT) in Mar 2009, with which he is currently an Associate Professor and PhD Supervisor, Dr. Xia was an Assistant Professor at Zhejiang University. He is on the Editorial Board of several international journals. He serves as General Chair, PC Chair, Workshop Chair, Publicity Chair, or PC Member of a number of conferences. He is also the Guest Editor of several journal special issues. Dr. Xia has (co-)authored one book and over 80 scientific papers. His research interests include cyber-physical systems (CPS), mobile computing, and social computing. He is a member of IEEE and ACM.

A Novel Differential Evolution with Uniform Design for Continuous Global Optimization

Lei Peng
School of Computer, China University of Geosciences, Wuhan, China
College of Computer Science, Huazhong University of Science and Technology, Wuhan,China
Email:penglei0114@gmail.com

Yuanzhen Wang
College of Computer Science, Huazhong University of Science and Technology, Wuhan,China
Email:wangyz2005@163.com

Guangming Dai
School of Computer, China University of Geosciences, Wuhan, China
Email: gmdai@cug.edu.cn

Zhongsheng Cao
College of Computer Science, Huazhong University of Science and Technology, Wuhan,China
Email: caozhongsheng@126.com

***Abstract*—Differential Evolution (DE) is a simple and efficient optimizer, especially for continuous global optimization. Over the last few decades, DE has often been employed for solving various engineering problems. At the same time, the DE structure has some limitations in the complicated problems. This fact has inspired many researchers to improve on DE by proposing modifications to the original algorithm. Population initialization is very important to the performance of differential evolution. A good initialization method can help in finding better solutions and improving convergence rate. In this paper, a uniform-differential evolution algorithm (UDE) is proposed. It incorporates uniform design initialization method into differential evolution to accelerate its convergence speed and improve the stability. UDE is compared with other four algorithms of Standard Differential Evolution (SDE), Orthogonal Differential Evolution (ODE), Opposition Based Differential Evolution(OBDE) and Chaos Differential Evolution(CDE). Experiments have been conducted on 23 benchmark problems of diverse complexities. The results indicate that our approach has the stronger ability and higher calculation accuracy to find better solutions than other four algorithms.**

***Index Terms*—differential evolution, global optimization, uniform design method, orthogonal design method, Opposition Based, Chaos Initialization**

I. INTRODUCTION

Global optimization is the task of finding the absolutely best set of parameters to optimize an objective function. Generally, there are solutions that are locally optimal but not globally optimal. Consequently, global optimization problems are typically quite difficult to solve exactly. Using classical determinate direct search techniques may fail to solve such problems because these problems usually contain multiple local optima.

The problem of finding a global minimum of the unconstrained optimization problem:

$$\min_{x \in R^n} f(x)$$

Where f is a generally nonconvex, real valued function defined on R^n.

In recent years, the use of alternative approaches to solve complex optimization problems is very common. Evolutionary Algorithms (EAs) such as genetic algorithm, evolutionary programming, evolution strategy and genetic programming have received many interests from researchers and practitioners due to their competitive results when solving this kind of problems.

Differential Evolution (DE) is a branch of evolutionary algorithms developed by Rainer Storn and Kenneth Price [1] for global continuous optimization problem. It has won the third place at the 1st International Contest on Evolutionary Computation. It shares similarities with previous EAs. For example, DE works with a population of solutions, called vectors, it uses recombination and mutation operators to generate new vectors and, finally, it has a replacement process to discard the less fit vectors. DE uses real encoding to represent solutions. Some of the differences with respect to other EAs are the following: DE uses a special mutation operator based on the linear combination of three individuals and a uniform crossover operator. It has several attractive features. Besides being an exceptionally simple evolutionary strategy, it is significantly faster and robust for solving numerical optimization problem and is more likely to find the functions true global optimum.

Despite having several striking features and successful applications to different fields, DE has sometimes been shown slow convergence and low accuracy of solutions

doi:10.4304/jcp.7.1.3-10

when the solution space is hard to explore. Many efforts have been made to improve the performance of DE and many variants of DE have been proposed.

The first direction for improvement is hybridization. Sun et al. [2] developed DE/EDA which combines DE with EDA for the global continuous optimization problem. It combines global information extracted by EDA with differential information obtained by DE to create promising solutions. The presented experimental results demonstrated that DE/EDA outperforms DE and EDA in terms of solution quality within a given number of objective function evaluations. Noman et al.[3] proposed a DE variant which incorporated a Local Search(LS) technique to solve optimization problem by adaptively adjusting the length of the search, using a hill-climbing heuristic. Experimenting with a wide range of benchmark functions,the results show that the proposed new version of DE performs better, or at least comparably,to classic DE algorithm. He et al.[4] proposed a new binary differential evolution algorithm based on the theory of immunity in biology. The test results show the improvement of the searching ability and increment in the convergence speed in comparison with the other algorithms. Das et al.[5]introduced a stochastic selection mechanism to improve the accuracy and convergence speed of DE. The idea of a conditional acceptance function (that allows accepting inferior solutions with a gradually decaying probability) is borrowed from the realm of the Simulated Annealing (SA). The resulting hybrid algorithm has been compared with three state-of-the-art adaptive DE schemes. The experiment results indicate that the mixed algorithm is able to find better solutions on a six-function testbed and one difficult engineering optimization problem. Omran et al.[6] incorporated a hybrid of concepts from chaotic search, opposition-based learning, differential evolution and quantum mechanics, named CODEQ to solve constrained problems. The experiment results indicate that CODEQ is able to find excellent solutions in all cases. Zhang et al.[7] proposed a hybrid of DE with PSO, called DE-PSO which incorporates concepts from DE and PSO, updating particles not only by DE operators but also by mechanisms of PSO. The presented experimental results demonstrate its effectiveness and efficiency. Wang et al.[8] combined the self-adaptive mixed distribution based univariate estimation of distribution algorithm (MUEDA) and a modified DE (MDE) to form a new algorithm, named ED-DE. It solved Economic Load Dispatch (ELD) problem successfully. Coelho et al.[9]combined ant colony optimization(ACO) with a differential evolution method (MACO) for chaotic synchronization. Jia et al.[10] proposed a Chaos and Gaussian local optimization based hybrid differential evolution (CGHDE) to high-dimensional complex engineering problems. The randomicity of chaotic local search can explore in a wide search space to overcome the premature in the earlier evolution phase and Gaussian optimization can refine the optimum in the later run phase. The experiment results indicate that CGHDE is able to find excellent solutions than other algorithms.

The second direction for improvement is dynamic adaptation of the control parameters. DE is sensitive to the two crucial parameters, to a certain extent the parameter values determine whether DE is capable of finding a near-optimum solution or not. So, recently, some studies focus on adaptive control parameters. Zaharie[11] proposed to transform F into a Gaussian random variable. Liu et al.[12] proposed a fuzzy adaptive differential evolution (FADE) which uses fuzzy logic controllers to adapt the mutation and crossover control parameters. Das et al. [13] proposed two schemes which are named DERSF and DETVSF to adapt the scaling factor F. Brest et al.[14] presented a novel approach to self-adapt parameters F and Cr. In their method, these two control parameters are encoded at the individual level. Nobakhti et al.[15] proposed a Randomised Adaptive Differential Evolution (RADE) method, which a simple randomised self-adaptive scheme is proposed for the DE mutation weighting factor F. Qin et al.[16] proposed self-adaptive DE (SaDE) which the trial vector generation strategies and two control parameters are dynamically adjusted based on their performance. Zhang et al.[17] proposed a new differential evolution (DE) algorithm (JADE) which the optional archive operation utilizes historical data to provide information of progress direction.Pan et al[18]proposed a self-adaptive DE algorithm, namely SspDE. It used an associated strategy list(SL),a mutation scaling factor F list (FL),and a crossover rate CR list (CRL) to be more effective in obtaining better quality solutions.

The third direction for improvement is population initialization. Before solving an optimization problem, it usually has no information about the location of the global minimum. It is desirable that an algorithm starts to explore those points that are scattered evenly in the decision space. Population initialization is a crucial task in evolutionary algorithms because it can affect the convergence speed and also the quality of the final solution. Recently, some researchers are working some methods to improve the EAs population initialization. Leung et al.[19] designed a GA called the orthogonal GA with quantization (OGA/Q) for global numerical optimization with continuous variables. Gong et al [20] used orthogonal design method to improve the initial population of DE(ODE). Rahnamayan et al. [21-23] proposed two novel initialization approaches which employ opposition-based learning and quasi-opposition to generate initial population. Xu et al.[24] used chaos initialization to get rapid convergence of DE as the region of global minimum. Pant et al.[25] proposed a novel initialization scheme called quadratic interpolation to DE with suitable mechanisms to improve its generation of initial population. Peng et al.[26] used Uniform-Quasi-Opposition to generate initial population of DE and accelerate its convergence speed and improve the stability. Ozer[27] used chaotic maps to generate sequences from different chaotic systems to construct initial population and proposed Chaotically Initialized Differential Evolution (CIDE).

In this paper, an improvement version of DE, namely Uniform-Differential Evolution (UDE) is presented to solve unconstrained optimization problem. UDE combines DE with uniform initialization. According to our previous study, uniform design generation can enhance the quality of initial population. The two experiments are designed and UDE is compared with SDE, ODE,OBDE,CDE. The experimental results show that UDE outperforms SDE, ODE, OBDE, CDE.

The paper is organized as follows: Section 2 provides an overview of differential evolution, uniform design method, orthogonal design method, opposition based method and Chaos initialization method. Our proposed approach is presented in detail in Section 3.After that, in Section 4 the experimental design, the results are included. The last section, Section 5, is devoted to conclusions and future works.

II. PRELIMINARY

A. Differential evolution

The DE algorithm in pseudo-code is shown in Algorithm 1. Each vector i in the population at generation t , x_i called target vector will generate one offspring called trial vector v_i . Trial solutions are generated by adding weighted difference vectors to the target vector. This process is referred to as the mutation operator where the target vector is mutated. A crossover step is then applied to produce an offspring which is only accepted if it improves on the fitness of the parent individual. Many variants of standard DE have been proposed, which use different learning strategies and/or recombination operations in the reproduction stage. In this paper, the DE/best/1/exp strategy is used.

Algorithm 1.Procedure of DE with best/1/exp

1: Generate the initial population P , define $x_i(t)$ as the i -th individual of the t -th generation:

$$x_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{in}(t))$$
$$i = 1, 2, \cdots, M; t = 1, 2, \cdots, t_{\max}$$

where n is the number of decision variable, M is the population size, $t_{\max}$ is the maximum generation.

2: Evaluate the fitness $f(x_i(t))$ for the each individual.
3: **while** the termination condition is not satisfied **do**
4: **for** i=1 to M **do**
5: Select x_{best} , x_{p1}, x_{p2} and $i \neq p1 \neq p2 \neq best$.
6: j=randint(1,n)
7: L=0
8: $v_i = P_i$
9: **repeat**
10: $v_{ij} = x_{bestj} + F \times (x_{p1j} - x_{p2j})$
11: j=(j+1) *mod n*
12: L=L+1
13: **until** $rand_{ij}$ [0,1)>CR *or* L>n
14: Evaluate the offspring v_i
15: **If** v_i is better than x_i **then**
16: $x_i = v_i$
17: **end if**
18: **end for**
19: **end while**
20: F is the scaling factor, CR is crossover factor.

B. Uniform design method

Experimental design method is a sophisticated branch of statistics. The uniform design, proposed by Fang and Wang[29] in 1980,is one of space filling designs and has been widely used in computer and industrial experiments. The main objective of uniform design is to sample a small set of points from a given set of points, such that the sampled points are uniformly scattered.

It defines the uniform array as $U_M q^n$, where n is factors and q is levels. When n and q are given, the population can be constructed by selecting M combinations from q^n .The steps of initialization population are as Algorithm 2.

Algorithm 2. Uniform Design Initialization

1: Find all the primer numbers $h = (h_1, h_2, \cdots, hs)$ which are less than M , where M is the size of population.
2: The j -th column of the uniform array is constructed according to (1)

$$U_{ij} = ih_j [\text{mod } M] \quad (1)$$
$$\text{where } i = 1, 2, \cdots, M \ ; \ j = 1, 2, \cdots, s$$

3: Suppose n ($n < s$) is the number of the variables, randomly choose $h_{i1}, \cdots, h_{jn}$ from the vector $h = (h_1, h_2, \cdots, hs)$. A uniform matrix of $U'_{M \times n}$ is constructed.
4: Generation of initial population

After constructing the uniform array, it can generate the uniform population which scatters uniformly over the feasible solution space according to (2).

$$P(i, j) = U^{'}_{ij} \times (u_j - l_j) / M + l_j$$
$$i = 1, 2, \cdots, M \ ; \ j = 1, 2, \cdots, n \quad (2)$$

where u_j and l_j are the maximum and minimum values of the variable j .

C. Orthogonal design method

Orthogonal design method [19,20] with both orthogonal array (OA) and factor analysis (such as the statistical optimal method) is developed to sample a small, but representative set of combinations for experimentation to obtain good combinations. OA is a fractional factorial array of numbers arranged in rows and columns, where each row represents the levels of factors in each combination, and each column represents a specific factor that can be changed from each combination. It can assure a balanced comparison of levels of any factor. The array is called orthogonal

because all columns can be evaluated independently of one another, and the main effect of one factor does not bother the estimation of the main effect of another factor.

Algorithm 3. Orthogonal Design Initialization

1: For k=1 to J do
2: $j = \frac{Q^{k-1}-1}{Q-1}+1$
3: For i=1 to R do
4: $a_{i,j} = \left\lfloor \frac{i-1}{Q^{j-k}} \right\rfloor \bmod Q$
5: End for
6: End for
7: For k=2 to J do
8: $j = \frac{Q^{k-1}-1}{Q-1}+1$
9: For s=1 to j-1 do
10: For t=1 to j-1 do
11: For i=1 to R do
12: $a_{i,(j+(s-1)(Q-1)+t)} = (a_{i,s} \times t + a_{i,j}) \bmod Q$
13: End for
14: End for
15:End for
16:End for
17:Increment $a_{i,j}$ by one for all $i \in [1,R]$ and $j \in [1,C]$
18:eval=0
19:For i=1 to R do
20: For j=1 to n do
21: k= $a_{i,j}$
22: $P(i,j) = l_j + (k-1)(\frac{u_j - l_j}{Q-1}), 1 \le k \le Q$
23: ($[l_j, u_j]$ is quantized Q-1 fractions)
24: End for
25:Evaluate $P(i,j)$ and eval++
26:End for
27:Sort the $P(i,j)$
28:Select the best M solution from $P(i,j)$ to generate the first population

D. Opposition Based Initialization

The concept of opposition-based learning (OBL) [21,22], in its earlier simple form, was introduced by Tizhoosh. The main idea behind OBL is the simultaneous consideration of an estimate and its corresponding opposite estimate in order to achieve a better approximation for the current candidate solution. As an advantage of opposite versus random points, purely random resampling or selection of solutions from a given population, has a higher chance of visiting or even revisiting unproductive regions of the search space.

Algorithm 4. Opposition Based Initialization

1: Generate uniformly distributed random population P_0
2: For i=0 to M do
3: For j=0 to n do
4: $OP_{0i,j} = a_j + b_j - P_{0i,j}$
5: Select M fittest individuals from set the $\{P_0, OP_0\}$ as the initial population.

E. Chaos Initialization

Chaos is a kind of characteristic of nonlinear systems and it has been extensively studied and applied in many fields[24,27]. Although it appears to be stochastic, it occurs in a deterministic nonlinear system under deterministic conditions. Chaotic sequences have been proven easy and fast to generate and store, there is no need for storage of long sequences. Merely a few functions (chaotic maps) and few parameters (initial conditions) are needed even for very long sequences. In addition, an enormous number of different sequences can be generated simply by changing its initial condition. Moreover, these sequences are deterministic and reproducible. Recently, chaotic sequences have been adopted instead of random sequences and very interesting and somewhat good results have been shown in many applications.

Algorithm 5. Chaos Initialization

1: Set the Maximum number if chaotic iteration, CI, according to the problem ,the population size M and i=0
2: While i≤M do
3: Randomly initialize chaotic variables taking into account the constrains , j=1,2,….,n and set counter k=0;
4: While (k<CI) do
5: Generate different chaotic variables cm_k^j , j=1,2,.,n, using Logistic map.
6: k=k+1
7: End While
8: Map the chaotic variables cm_k^j to feasible region according to equation $X_{j,i}^0 = X_j^{\min} + cm_k^j \times (X_j^{\max} - X_j^{\min})$, j=1,2,.,n
9: Set i=i+1
10:End While

III. UNIFORM DIFFERENTIAL EVOLUTION

The performance of DE is sensitive to the choice of control parameters. Based on our former research, the better choice of the parameters are $F = 0.5$ and $CR = 0.9$ .In order to avoid tuning the parameter F and CR , a parameter control technology is adopted according to the following scheme:

$$F = N(0.5, 0.02),\ CR = N(0.9, 0.02) \quad (3)$$

$N(\tau, \varepsilon)$ is a normal distribution that can generate values in the range of $[\tau - 3\times\varepsilon, \tau + 3\times\varepsilon]$.

Algorithm 6. Main procedure of Uniform Differential Evolution

1: **Initialization:** construct the population P by uniform initialization.

2: **Optimization using DE with Best/1/Exp.**

- **Mutation**

Select the best individual x_{best} in the t -th generation and two different individuals x_{p1}, x_{p2} from population where $i \neq p1 \neq p2 \neq best$.

- **Crossover**

Crossover operation is used to increase the diversity ,

- **Selection**

Compare $v_i(t)$ with $x_i(t)$, select the vector which has a better fitness as the individual in the new generation :

3: If stop criterion is met, go to step 4,else go to step 2

4: **Terminate**

IV. EXPERIMENTS

In order to assess the performance of our proposed algorithm, a comprehensive set of benchmark functions, including 23 different global optimization problems f01~f23 [20,28],have been employed for performance verification of the proposed approach. The formal definitions of the test functions and their global optimum(s) are summarized in [30]. Generally, following characteristics are desirable to provide a comprehensive test suite:

Functions 1 ~ 5 are unimodal problems and functions 8~ 13 are multimodal. Functions 6~7 are two special problems exhibiting a step landscape and a noisy landscape respectively. Functions 14~23 are low-dimensional functions which have only a few local minima.

Two experiments are designed. For each test functions, it performs 50 independent runs for each algorithm with different random seeds.

The first test compares the convergence speed of UDE with SDE, ODE,OBDE,CDE by measuring the number of successful runs and the mean number of function calls (NFC) of successful runs which are the most commonly used metrics. The test results of SDE and ODE come from the literature [20].

In the first experiment, the parameters of UDE are as follows:

- Population Size: NP=100.
- Maximum number of NFC(MAX_{NFC}) is 500000.
- The scaling factor F and probability of crossover CR of UDE use parameter control scheme as (3) .
- Stopping criterions are

$|f(x_{best}) - f(x_{optimal})| \leq 0.005$ or MAX_{NFC} is reached,

where $f(x_{best})$ is the best solution in the current run,

$f(x_{optimal})$ is the globally minimal function value.

The results are list in Table Ⅰ. From this table it can firstly be observed that ODE,OBDE and UDE can solve 23 benchmark problems in all 50 runs, but SDE cannot solve function f05 and f07 in all runs and it traps in the local optima once and four times. CDE traps in the local optima six times on function f20.Secondly, UDE needs less mean NFEs of successful runs than SDE, ODE, OBDE,CDE in 17 test functions f02~f06,f10~f13, f15~f17 and f19~f23. Especially in functions f04,f05, f06,f20,f21,f22,f23, it can be found that UDE makes considerable reduction of the mean NFEs of successful runs .

From these discussions, it can be concluded that firstly, the performance of UDE is better than other four algorithms; secondly, the uniform design can accelerate DE's convergence speed.

The second experiment compares the stability and calculation accuracy among the five algorithms. UDE has been compared with SDE, ODE,OBDE,CDE. The performance metrics have: (1)the mean NFEs(MNFEs) (2) the mean best function value(Mean best) (3)the standard deviation of the function values(Std).It performs 50 independent runs for each algorithm on the benchmark problems.

The parameters of UDE are as follows:

- Population Size: NP=100
- Maximum number of function calls is on Table Ⅱ
- The scaling factor F and probability of crossover CR of UDE use parameter control scheme as (3)

The mean results of 50 independent runs are summarized in Table Ⅱ. Results for SDE, ODE are taken from [20]. From Table Ⅱ, it can be seen that UDE needs less function evaluations than SDE, ODE,OBDE,CDE in 6 functions(f06, f09, f11, f14, f15, f16). UDE can provide better mean best results than SDE, ODE,OBDE,CDE for 7 functions (f03, f07, f10, f12, f13, f15, f20). Furthermore, UDE obtains smaller standard deviation than other four algorithms in 10 functions (f01, f03,f04,f10,f12, f13, f20, f21,f22,f23).

The results of the mean function values indicate that UDE is able to obtain more accurate solutions .The results of the standard deviation of the function values present that UDE is more stable than other four algorithms. Also, these results demonstrate that uniform design initialization used in DE can be effectively worked and enhance the performance of DE and accelerate the convergence speed and improve the stability and calculation accuracy of differential evolution.

V. CONCLUSIONS

In this article, it has presented a new variant of differential evolution algorithm (UDE) in which the initial population is selected using the uniform design initialization method. An adaptive parameter control technology is adopted .UDE has compared with other four algorithms of SDE, ODE,OBDE,CDE. According to the experiment results, it can conclude that uniform design initialization can enhance the capability of our algorithm and UDE is better and more stable than other four algorithms on the benchmark problems.

Future work consists on extending the present version for solving some real life optimization problems and combining uniform differential evolution with other local optimizer.

TABLE I. COMPARISON WITH SDE, ODE,OBDE,CDE AND UDE ON $f1-f23$. THE BETTER RESULTS OF NUMBER OF SUCCESSFUL RUNS AND MEAN NFES OF SUCCESSFUL RUNS IN **BOLDFACE**.

F	Number of successful runs					Mean NFEs of successful runs				
	SDE	ODE	OBDE	CDE	UDE	SDE	ODE	OBDE	CDE	UDE
f01	50	50	50	50	50	53548	35235	**16752**	17274	16867
f02	50	50	50	50	50	45912	36914	21344	21698	**20547**
f03	50	50	50	50	50	144076	95520	47110	47720	**46991**
f04	50	50	50	50	50	189680	126731	188408	199598	**111897**
f05	49	50	50	50	50	236808	232171	310538	315624	**227266**
f06	50	50	50	50	50	30286	20051	35816	36984	**17521**
f07	46	50	50	50	50	328491	**83072**	161828	176882	129778
f08	50	50	50	50	50	95590	**42346**	81452	79596	101567
f09	50	50	50	50	50	168732	**63763**	194664	196318	77927
f10	50	50	50	50	50	53784	36802	63386	64768	**35341**
f11	50	50	50	50	50	51602	34010	59810	61448	**32952**
f12	50	50	50	50	50	36290	23648	45312	46270	**23595**
f13	50	50	50	50	50	50236	33409	50770	51794	**21949**
f14	50	50	50	50	50	3702	3383	**412**	844	525
f15	50	50	50	50	50	946	1124	1076	1458	**868**
f16	50	50	50	50	50	998	1016	642	868	**606**
f17	50	50	50	50	50	1356	1584	562	894	**454**
f18	50	50	50	50	50	1556	1621	**800**	1054	808
f19	50	50	50	50	50	1038	946	520	780	**404**
f20	50	50	50	44	50	14504	4059	3278	4634	**1193**
f21	50	50	50	50	50	5918	5473	5650	6440	**4141**
f22	50	50	50	50	50	5066	5053	5444	6548	**1959**
f23	50	50	50	50	50	5184	4782	1614	6332	**1547**

ACKNOWLEDGMENT

The authors would like to acknowledge the anonymous reviewers for their useful comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China under Grant No.60873107, the National High-Tech Research and Development Plan of China under Grant No. 2008AA12A201,the Fundamental Research Funds for the Central Universities under Grant No. CUGL090238 and CUG100708, the Research Foundation for Outstanding Young Teachers China University of Geosciences (Wuhan) under Grant No. CUGQNL0831.

REFERENCES

[1] R. Storn and K. Price, “Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces”,Journal of Global Optimization., vol. 11, pp. 341-359, 1997.

[2] J. Sun, Q. Zhang and E. P. K. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," Information Sciences, vol. 169, pp. 249-262, 2005.

[3] N. Noman and H. Iba, "A new generation alternation model for differential evolution," in 8th Annual Genetic and Evolutionary Computation Conference 2006, July 8, 2006 - July 12, 2006, Seattle, WA, United states, 2006, pp. 1265-1272.

[4] X. He and L. Han, "A novel binary differential evolution algorithm based on artificial immune system," in 2007 IEEE Congress on Evolutionary Computation, CEC 2007, September 25, 2007 - September 28, 2007, Singapore, pp. 2267-2272.

[5] S. Das, A. Konar and U. K. Chakraborty, "Annealed differential evolution," in 2007 IEEE Congress on Evolutionary Computation, CEC 2007, September 25, 2007 - September 28, 2007, Singapore, pp. 1926-1933.

[6] M. G. H. Omran and A. Salman, "Constrained optimization using CODEQ," Chaos, Solitons and Fractals, vol. 42, pp. 662-668, 2009.

[7] C. Zhang, J. Ning, S. Lu, D. Ouyang, and T. Ding, "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization," Operations Research Letters, vol. 37, pp. 117-122, 2009.

[8] Y. Wang, B. Li and T. Weise, "Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems," Information Sciences, vol. 180, pp. 2405-2420, 2010.

[9] L. D. S. Coelho and D. L. D. A. Bernert, "A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization," Expert Systems with Applications, vol. 37, pp. 4198-4203, 2010.
[10] D. Jia and G. Zheng, "Hybrid differential evolution combined with chaos and Gaussian local optimization," Kongzhi yu Juece/Control and Decision, vol. 25, pp. 899-902, 2010.in Chinese
[11] D.Zaharie, "Critical values for the control parameters of differential evolution algorithms", in Proc. MENDEL 8th Int.Conf.Soft Comput.,2002,pp.62-67
[12] J.Liu and J.Lampinen, "A fuzzy adaptive differential evolution algorithm", Soft Computing--A Fusion of Foundations, Methodologies and Applications,vol.9,no.6, pp.448-642, 2005.
[13] S. Das, A. Konar and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in GECCO 2005 - Genetic and Evolutionary Computation Conference, June 25, 2005 - June 29, 2005, Washington, D.C., United states, 2005, pp. 991-998.
[14] J. Brest, S. Greiner, B. Bokovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," IEEE Transactions on Evolutionary Computation, vol. 10, pp. 646-657, 2006.
[15] A. Nobakhti and H. Wang, "A simple self-adaptive Differential Evolution algorithm with application on the ALSTOM gasifier," Applied Soft Computing Journal, vol. 8, pp. 350-370, 2008.
[16] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in 2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005, September 2, 2005 - September 5, 2005, Edinburgh, Scotland, United kingdom, 2005, pp. 1785-1791.
[17] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," IEEE Transactions on Evolutionary Computation, vol. 13, pp. 945-958, 2009.
[18] Q. Pan, P. N. Suganthan, L. Wang, L. Gao, and R. Mallipeddi, "A differential evolution algorithm with self-adapting strategy and control parameters," Computers & Operations Research, vol. 38, pp. 394-408, 2011.
[19] Y. W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," IEEE Transactions on Evolutionary Computation, vol. 5, pp. 41-53, 2001.
[20] W. Gong, Z. Cai and L. Jiang, "Enhancing the performance of differential evolution using orthogonal design method," Applied Mathematics and Computation, vol. 206, pp. 56-69, 2008.
[21] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "A novel population initialization method for accelerating evolutionary algorithms," Computers & Mathematics with Applications, vol. 53, pp. 1605-1614, 2007.
[22] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Opposition-based differential evolution," IEEE Transactions on Evolutionary Computation, vol. 12, pp. 64-79, 2008.
[23] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Quasi-oppositional differential evolution," in 2007 IEEE Congress on Evolutionary Computation, CEC 2007, September 25, 2007 - September 28, 2007, Singapore, 2008, pp. 2229-2236.
[24] X.Xu, X.Huang, and D.Qain, "Adaptive accelerating differential evolution," Complex Systems and Complexity Science ,5(3),pp:87-92,2008. In Chinese
[25] M. Pant, M. Ali and V. P. Singh, "Differential evolution using quadratic interpolation for initializing the population," in 2009 IEEE International Advance Computing Conference, IACC 2009, March 6, 2009 - March 7, 2009, Patiala, India, 2009, pp. 375-380.
[26] L. Peng and Y. Wang, "Differential evolution using uniform-quasi-opposition for initializing the population," Information Technology Journal, vol. 9, pp. 1629-1634, 2010.
[27] A. Bedri Ozer, "CIDE: chaotically initialized differential evolution," Expert Systems with Applications, vol. 37, pp. 4632-4641, 2010.
[28] X. Yao, Y. Liu and G. Lin, "Evolutionary programming made faster," IEEE Transactions on Evolutionary Computation, vol. 3, pp. 82-102, 1999.
[29] Y. Wang and K. T. Fang, "A note on uniform distribution and experimental design", KEXUE TONGBAO, vol. 26, no. 6, pp. 485-489,1981.In Chinese.
[30] L.Peng,Y.Wang,and G.Dai, "UDE:differential evolution with uniform design",in 3rd International Symposium on Parallel Architectures,Algorithms and Programming, PAAP 2010, December 18, 2010-December 20, 2010, Dalian, China, 2010, pp.239-246

Lei Peng is a lecturer in School of Computer Science in China University of Geosciences, Wuhan, China and a Ph.D. student at College of Computer Science and Technology, Huazhong University of Science and Technology. The research interests are evolutionary algorithm and evolutionary engineering optimal design.

Yuanzhen Wang is Professor in College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. The research interest is DataBase.

Guangming Dai is Professor in School of Computer Science in China University of Geosciences, China. His main interests are in the area of evolutionary algorithm and mission analysis and design.

Zhongsheng Cao is Associate Professor in College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. The research interest is DataBase and Data Mining.

TABLE II. COMPARISION WITH DE, ODE, OBDE,CDE,UDE. MAX_EVAL : THE MAX NUMBER OF FUNCTION CALLS, MEAN BEST: THE MEAN BEST FUNCTION VALUE (OVER 50 TRIALS), STD: THE STANDARD DEVIATION OF THE FUNCTION VALUES. THE BEST MNFES, MEAN BEST AND STD IN **BOLDFACE**.

F	Max_eval	**MNFEs**					**Mean best**					**Std**				
		SDE	ODE	OBDE	CDE	UDE	SDE	ODE	OBDE	CDE	UDE	SDE	ODE	OBDE	CDE	UDE
f01	150000	150000	150000	**127952**	128696	128653	1.64E-18	2.06E-23	**8.91E-51**	9.11E-51	8.97E-51	5.29E-18	1.83E-23	8.28E-52	8.29E-52	**6.69E-52**
f02	200000	200000	200000	200000	200000	200000	2.97E-15	1.43E-18	**1.83E-42**	2.83E-42	4.55E-42	5.78E-15	8.11E-19	**1.64E-42**	2.87E-42	4.87E-42
f03	500000	500000	500000	**362474**	365588	366015	3.53E-20	5.25E-27	9.47E-51	9.41E-51	**9.36E-51**	3.53E-20	9.66E-27	5.07E-52	5.64E-52	**4.80E-52**
f04	500000	500000	500000	500000	500000	500000	9.73E-10	**2.72E-15**	2.22E-07	2.11E-07	2.74E-15	5.00E-10	9.30E-15	8.14E-07	7.19E-07	**5.17E-15**
f05	500000	494788	**428776**	500000	500000	486784	2.55E-29	**0**	1.23E-16	7.29E-18	1.63E-27	1.15E-28	**0**	4.26E-16	1.06E-17	7.97E-27
f06	150000	30454	22640	36390	36798	**17539**	0	0	0	0	0	0	0	0	0	0
f07	300000	300000	300000	300000	300000	300000	0.00598	0.00145	0.00279	0.00288	**0.00137**	0.00125	**4.20E-04**	9.92E-04	1.38E-03	5.52E-04
f08	300000	167324	**90381**	300000	300000	131610	-12569.48662	-12569.48662	-12569.48662	-12569.48662	-12569.48662	0	0	7.35E-12	7.35E-12	**0**
f09	300000	247626	127666	286512	288890	**125567**	0	0	0	7.11E-17	0	0	0	0	5.02E-16	0
f10	150000	150000	150000	150000	150000	150000	3.19E-10	4.67E-13	1.52E-08	1.74E-08	**4.64E-15**	1.10E-10	1.86E-13	3.47E-09	3.47E-09	**1.25E-15**
f11	200000	138236	109853	162874	162866	**88541**	0	0	0	0	0	0	0	0	0	0
f12	150000	150000	150000	150000	150000	150000	4.99E-20	6.73E-26	1.89E-16	2.22E-16	**1.57E-32**	3.68E-20	9.27E-26	9.12E-17	8.83E-17	**1.66E-47**
f13	150000	150000	150000	150000	150000	150000	4.42E-18	4.37E-24	1.07E-15	1.33E-15	**1.35E-32**	4.66E-18	3.67E-24	4.29E-16	6.07E-16	**8.29E-48**
f14	10000	9796	9552	2220	2436	**2079**	0.998	0.998	0.998	0.998	0.998	7.92E-15	**0**	1.01E-15	1.01E-15	1.01E-15
f15	150000	34484	32430	150000	150000	**18150**	3.08E-04	3.08E-04	3.075E-04	3.99E-04	**3.075E-04**	0	**0**	2.14E-19	2.77E-04	6.45E-19
f16	10000	10000	10000	10000	6894	**2871**	-1.03163	-1.03163	-1.03163	-1.03163	-1.0316285	9.16E-14	**0**	3.09E-16	6.13E-16	6.73E-16
f17	10000	10000	10000	**2868**	3232	10000	0.39789	0.39789	0.39789	0.39789	0.39789	6.35E-11	2.01E-10	3.36E-16	3.36E-16	**3.36E-16**
f18	10000	10000	10000	10000	10000	10000	3	3	3	3	3	1.34E-14	**0**	3.22E-015	2.69E-15	3.56E-15
f19	10000	10000	10000	10000	10000	10000	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	2.68E-15	**2.68E-15**	2.69E-15	2.69E-15	2.69E-15
f20	20000	20000	20000	20000	20000	20000	-3.31962	**-3.322**	-3.279	-3.244	**-3.322**	0.01681	1.13E-12	5.76E-02	5.69E-02	**3.17E-16**
f21	10000	10000	10000	10000	10000	10000	-10.1532	-10.1532	-10.1532	-10.1532	-10.1532	1.29E-05	1.04E-06	3.07E-06	1.33E-06	**7.76E-15**
f22	10000	10000	10000	10000	10000	10000	-10.40294	-10.40294	-10.40294	-10.40294	-10.40294	5.84E-08	2.49E-08	5.99E-07	1.48E-05	**3.30E-10**
f23	10000	10000	10000	10000	10000	10000	-10.53641	-10.53641	-10.53641	-10.53641	-10.53641	5.80E-08	2.35E-08	2.04E-07	1.201E-05	**8.44E-11**

Leveraging 1-hop Neighborhood Knowledge for Connected Dominating Set in Wireless Sensor Networks

Wenyong Wang
School of Computer Science and Engineering in University of Electronic Science and Technology of China, Chengdu, China
Email: wangwy@uestc.edu.cn

Jun Zhang, Yong Tang, Yu Xiang and Ting Yang
School of Computer Science and Engineering in University of Electronic Science and Technology of China, Chengdu, China
Email: jun.wudu@gmail.com, {worldgulit, jcxiang, yting}@uestc.edu.cn

***Abstract*—To improve the efficiency of routing and broadcast and reducing energy consumption in the process of data transmission, calculating minimum connected dominating set is always used to construct virtual backbone network in wireless sensor networks. Calculating the minimum connected dominating set (MCDS) of plane graphs is a NP-complete problem. In this paper, an algorithm leveraging 1-hop neighborhood knowledge for connected dominating set is proposed. First, the minimum forwarding set is calculated severally by each node in the entire network. Then any one node can start the process of broadcasting messages including the information of minimum forwarding set in the network. Finally, the connected dominating set of the entire network is achieved by exchanging information. The proposed algorithm aims to get a small connected dominating set, meanwhile, to minimize the consumption of energy and time. The simulation results show that the algorithm has achieved its purpose with fast convergence, low transmission traffic and reasonable size of connected dominating set.**

***Index Terms*—Wireless Sensor Networks, 1-hop, Communication Coverage, Connected Dominating Set**

I. INTRODUCTION

Constructing virtual backbone network in wireless sensor networks has a wonderful performance in improving the performance of broadcast [1], reducing reduplicate data transmission, energy saving and bandwidth saving. Constructing a virtual backbone is identical with calculating connected dominating set in graph theory [2]. The virtual backbone of the network derived from a smaller connected dominating set [3]will not only benefit the design of energy-efficient routing, but also save energy for the reason that non-dominating nodes without monitoring task could enter sleep (energy-saving) mode.

Manuscript received December 12, 2010; revised March 22, 2011; accepted April 3, 2011.

This paper is supported by Research Fund for the Doctoral Program of Higher Education of China for New Teachers No.200806141110 and project CNGI of NDR No.CNGI-09-01-07.

As calculating the minimum connected dominating set (MCDS) of arbitrary graphs is a NP-complete problem, heuristic algorithms are usually used to calculate it approximately. There are two main heuristic algorithms: centralized algorithm [4-6] and distributed algorithm [7-15]. Although the minimum connected dominating set can be calculated by centralized algorithms, it is difficult to get the entire topology structure of dynamic wireless sensor networks which is necessary for centralized algorithms. Hence distributed algorithm is adopted in this paper. Typical distributed algorithms, such as Dai's algorithm [13] and MISB algorithm [14] need two or more hops adjacent node information to calculate the minimum connected dominating set. As a result, more messages are exchanged for calculating the minimum connected dominating set, leading to more energy consumption and time delay.

To reduce energy consumption and message exchange, improve the efficiency of algorithm and get a smaller connected dominating set, a new algorithm, Leveraging 1-hop Neighborhood Knowledge for Connected Dominating Set in Wireless Sensor Networks (OHCDS), which is based on our earlier achievement OHDC algorithm [15], is proposed in this paper. There are two stages in the calculating process of our algorithm OHCDS. In the first stage each node calculates the minimum forwarding set severally. In the second stage, the node checks out if it is a dominating node by broadcasting messages including the information of minimum forwarding set, the process of which is similar to breadth-first search of graph.

The rest of this paper is organized as follows. Section 2 describes symbols used in this paper. Section 3 analyses the CDS problem from another view. Section 4 presents details of our algorithm. Section 5 presents performance evaluation. Section 6 concludes the paper.

II. SYMBOLS DESCRIPTION

Symbols used in this paper are shown in Tab.1. We assume that transmission radius of all nodes is *R*, and all

doi:10.4304/jcp.7.1.11-18

nodes are randomly dispersed over a 2-dimensional geographical region. The communication region of the node v is $A(v)$. The whole wireless sensor network region is regarded as a connected graph G with V representing set of vertices and E representing set of edges. We get $uv \in E$ if and only if vertices u, v are adjacent and $u, v \in V$.

TABLE 1. SYMBOL TABLE

Symbol	definition
G	Equation $G=(V,E)$ denotes a graph G, V is the vertices set, and E is the edges set
R	The transmission radius of node
$N(v)$	The adjacent node set of vertex v
$N[v]$	$N[v]=N(v)\cup\{v\}$
$C(v)$	The circle with v as the center and a radius of R
$A(v)$	The region of circle $C(v)$
$A(D)$	$A(D)=\bigcup_{v\in D}A(v)$, where D is a set of vertices
$Q(v)$	The set of $A(N[v])$'s boundary intersection points
$B(v)$	If $v'\in B(v)$, then $C(v')$ is a circle at $A(N[v])$'s boundary
$\overset{\frown}{ab}$	Arc $\overset{\frown}{ab}$, whose direction is clockwise from a to b
uv	The distance between node u and node v, $u,v\in V$
CDS	The connected dominating set of graph $G=(V,E)$

III. PROBLEM STATEMENT

If source node s broadcasts packet P, and all nodes in $N(s)$, which is the adjacent node of s, forward this packet after they receive it, then all 2-hop adjacent nodes of s can receive the packet P. If we can obtain a forwarding set $F(.)$ which satisfies $A((F(s)\cup\{s\}))=A(N[s])$ $(F(s)\subseteq N(s))$, then all 2-hop adjacent nodes of s can receive the packet P as long as all the nodes in $F(s)$ participate in forwarding. If all the nodes in the network calculate their own forwarding set $F(.)$, and only nodes in $F(.)$ are designated to forward this packet P, then all the nodes can receive the packet P. These nodes which forward the packet form the connected dominating set CDS.

In order to obtain a smaller CDS, first, this algorithm makes use of local communication coverage information to calculate the minimum forwarding set $F_{\min}(.)$ of every vertex (*algorithm* 1 in section IV). Then, choose a source vertex s from V, and finally we can obtain an ideal CDS by broadcasting (*algorithm* 2 in section IV).

***Lemma* 1.**

①. $A(B(v)\cup\{v\})=A(N[v])$, ②. $F_{\min}(v)=B(v)$.

***Proof*:**

①. The boundary of region $A(N[v])$ is made up of the arcs whose center forms the set $B(v)$. Assume that $A(N[v])-A(B(v)\cup\{v\})\neq\varnothing$, then, the hole formed is inside region $A(N[v])$.

Assume that vertex c is a point inside the hole. If connect vertex c and vertex v, ray vc intersects circle $C(v')$ which is at the boundary of the hole at point a (a is out of line segment $\overline{vc}$). Since line segment $\overline{vv'}\le R$, vertex v is inside or on the circle $C(v')$. Moreover, point a is on the circle $C(v')$. Then line segment $\overline{va}$ is inside $C(v')$. So each point of $\overline{va}$ is inside or on circle $C(v')$. c is a point of $\overline{va}$ means that c is inside circle $C(v')$ as well, hence point c is not inside that hole which contradicts the assumption. So the assumption is not true. So there is no hole in $A(N[v])$, and $A(B(v)\cup\{v\})=A(N[v])$ is gotten.

②. We prove by the method of reduction to absurdity. From ① of *lemma* 1, we know that $F_{\min}(v)$ must satisfy $|F_{\min}(v)|\le|B(v)|$. Assume that $F_{\min}(v)\neq B(v)$. Since $A(B(v)\cup\{v\})=A(N[v])$, we get $A(F_{\min}(v)\cup\{v\})\neq A(N[v])$ or $A(F_{\min}(v)\cup\{v\})=A(N[v])$ and $F_{\min}(v)\supset B(v)$ $(|F_{\min}(v)|>|B(v)|)$. Those two results contradict the preconditions, so $F_{\min}(v)=B(v)$.

■

From ② of *lemma* 1, we know the required $F_{\min}(v)$ is identical with $B(v)$. For convenience, we divide the CDS problem of graph G into two problems:

Problem A is how to calculate set $B(v_x)$ for each vertex $v_x\in V$ of graph G.

Problem B is how to calculate CDS of graph G start from source s.

IV. ALGORITHM IMPLEMENTATION

A Solving Problem A

Assume $d_k=\{(x_k,y_k);(u_k,v_k)\}\in Q(v_x)$ with coordinate (x_k,y_k), and it is the intersection point of circle $C(u_k)$ and circle $C(v_k)$, which are both at the boundary of region $A(N[v_x])$ (That is, $u_k,v_k\in B(v_x)$). $B(v_x)$ can be obtained by calculating $Q(v_x)$.

Here we introduce a method to calculate $Q(v_x)$. If $u_1,u_2,...,u_i\in N(v_x)$ and $u_1v_x<u_2v_x<...<u_iv_x$, then $C(u_1)$, $C(u_2)$, … , $C(u_i)$ will be added to the obtained region in turn.

In this algorithm, let v_x to be a reference vertex. The gray region in Fig.1 represents current region $A(P(u_i))$. We use u_0 to replace v_x, let the algorithm start from u_0. Assume $u_1,u_2,...,u_i\in N(v_x)$ and $u_1v_x<u_2v_x<...<u_iv_x$, thus we get current region $A(P(u_i))$ satisfies $A(P(u_i))=\bigcup_{j=0}^{i}A(u_j)$, in which $P(u_i)=\bigcup_{j=0}^{i}u_j$.

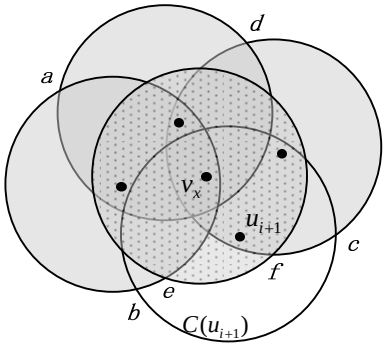


Figure 1. The process of adding circles

In Fig.1, $Q(v_x)=\{a,e,f,d\}$. $Q(v_x)$ will be updated and current region $A(P(u_i))$ will be expanded to $A(P(u_{i+1}))=A(P(u_i))\bigcup A(u_{i+1})$, if $u_{i+1}\in N(v_x)$ and circle $C(u_{i+1})$ is added to $A(P(u_i))$, then region $A(u_{i+1})$ covers vertices *e* and *f* in $Q(v_x)$, so vertices *e* and *f* are no longer at the boundary of $A(P(u_{i+1}))$. Thus *e* and *f* can be deleted from $Q(v_x)$. Simultaneously, new points *b* and *c*, the intersection points of $C(u_{i+1})$ and original region, are also at the boundary of new region. Then *b* and *c* are added into set $Q(v_x)$, and $Q(v_x)=\{a,b,c,d\}$.

According to the thought above, each time a vertex u is picked from $N(v_x)$, and $C(u)$ is added to existing region $A(P(u_i))$. If $A(u)$ covers existing vertices in $Q(v_x)$, those vertices should be deleted from $Q(v_x)$. If added $C(u)$ brings new intersection points with boundary, those new points are added into set $Q(v_x)$. This process continues until all vertices in $N(v_x)$ are added. Thus region $A(N[v_x])$ is formed, and the set of boundary intersection points $Q(v_x)$ is gotten. The set $Q(v_x)$ is just we wanted.

The correctness of this algorithm is ensured by *lemma* 2.

***Lemma* 2.**

Before adding $C(u_{i+1})$, there is no hole in the already formed region $A(P(u_i))$ which contents i added circles and $C(v_x)$.

***Proof* :**

lemma 2 can be proved by *lemma* 1 easily.

■

For convenience, the whole algorithm is divided into main algorithm and sub-algorithm.

In this algorithm, $P(u_i)$ is the set to store vertices that already added and $Q(u_i)$ is the set to store boundary intersection points of already-existed region.

***Algorithm* 1:** Calculating $B(v_x)$ of v_x

Input: set $N[v_x]$.

Output: set $B(v_x)$.

1) Initialize $A(P(u_0))=\varnothing$, $P(u_0)=\varnothing$ and $Q(u_0)=\varnothing$.

2) Put v_x into set $P(u_0)$. Add $C(v_x)$ to $A(P(u_0))$, and let $A(P(u_0))=A(v_x)$.

3) If v_x, u_1, u_2, … , u_i have been put into set P, then let $A(P(u_i))=A(v_x)\cup A(u_1)\cup A(u_2)\cup ...\cup A(u_i)$. Choose vertex u_{i+1} which is the nearest point to v_x in $N(v_x)-P(u_i)$.

4) Add $C(u_{i+1})$ to region $A(P(u_i))$, and $A(P(u_{i+1}))=A(P(u_i))\cup A(u_{i+1})$.

5) If $A(u_{i+1})$ covers some vertices in $Q(u_i)$, calculate set $Q(u_{i+1})$ by *sub-algorithm*, and then go to step 8).

6) If there is no vertex in $Q(u_i)$ covered by set $A(u_{i+1})$, then choose the nearest vertex u_j to u_{i+1} from $P(u_i)$.

7) Calculate the intersection points of circle $C(u_j)$ and circle $C(u_{i+1})$, and then put them into set $Q(u_i)$.

8) Let $i=i+1$, return step 3) until $N(v_x)-P(u_i)=\varnothing$.

9) For each $d_k\in Q$, put u_k, v_k associated with d_k into set $B(v_x)$.

10) End.

***Theorem* 1.** According to the step 3) of the *algorithm* 1,

a. adding a circle $C(u_{i+1})$ to the current region $A(P(u_i))$ will always update set $Q(v_x)$

b. added circle $C(u_{i+1})$ will intersect the boundary of region $A(P(u_i))$ at two and only two points .

***Proof* :**

(a) From step 3), it is known that, for $u_{i+1}\in N(v_x)-P(u_i)$, to $\forall u_k\in P(u_i)$, satisfies $u_{i+1}v_x>u_kv_x$, k=1,2,…,i. Thus by adding circle $C(u_{i+1})$ to $A(P(u_i))$, the circle $C(u_{i+1})$ can always intersects the boundary of $A(P(u_i))$. So set $Q(v_x)$ can always be updated.

(b) We use broken circle S' to denote the maximum scope $A(P(u_i))$ can reach. There are only two situations when adding circle $C(u_{i+1})$: ①circle $C(u_{i+1})$ intersects with circle S'; ②circle $C(u_{i+1})$ is tangent with circle S'. Fig.2 shows situation①. The gray part represents communication region of reference vertex v_x, and circle $C(u_{i+1})$ is the circle to be added. Circle *B* is in the current region. Circle $C(u_{i+1})$ intersects circle $C(v_x)$ at points *a* and *b*, and intersects circle S' at points *c* and *d*.

Assume that arc $\overset{\frown}{de}$ is the part of circle $C(u_{i+1})$ cut by circle *B* and circle $C(v_x)$, point *a* is the intersection point of circle $C(u_{i+1})$ and circle $C(v_x)$, point *e* is the intersection point of circle $C(u_{i+1})$ and circle *B*, arc $\overset{\frown}{de}$ is arc of circle *B*. If point *e* at the boundary of the current region $A(P(u_i))$, then any circle in current region $A(P(u_i))$

intersects arc $\overset{\frown}{ac}$ at a point on arc $\overset{\frown}{ae}$, thus point e is the only point on arc $\overset{\frown}{ac}$ at the boundary of region $A(P(u_i))$. Similarly, there is one and only one point of arc $\overset{\frown}{db}$ at the boundary of $A(P(u_i))$. As arc $\overset{\frown}{ba}$ is arc of circle $C(v_x)$, that is, it is inside current region $A(P(u_i))$, then, it has no intersection point with the boundary of $A(P(u_i))$. So does arc $\overset{\frown}{cd}$ due to being out of circle S'. Hence circle $C(u_{i+1})$ has two and only two intersection points with the current region $A(P(u_i))$. The situation ② is equal to situation ① when there is only one intersection point.

■

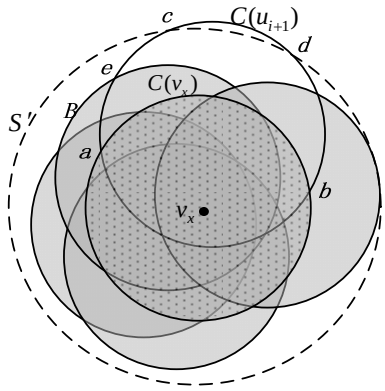


Figure 2. $C(u_{i+1})$ intersects with circle S'

Step 6) and step 7) of *algorithm*1 deal with the situation that there is no vertex in $Q(v_x)$ covered by region $A(u_{i+1})$. That is to find the nearest vertex u_j to u_{i+1} from $P(u_i)$ and calculate the intersection points of circle $C(u_j)$ and circle $C(u_{i+1})$, then add them into set $Q(v_x)$. So $Q(v_x)$ is updated. The correctness of this algorithm is ensured by the *theorem* 2 followed.

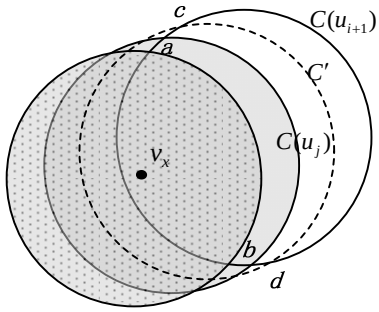


Figure 3. $C(u_{i+1})$ does not cover the intersection points of boundary

***Theorem* 2.** If $A(u_{i+1})$ does not cover any vertices in $Q(v_x)$ after adding circle $C(u_{i+1})$, and circle $C(u_j)$ is the nearest circle in $A(P(u_i))$ to circle $C(u_{i+1})$, then the two intersection points of circle and circle $C(u_{i+1})$ are at the boundary of $A(P(u_i))$.

***Proof* :**

As shown in Fig.3, the gray part represents $A(P(u_i))$. Prove by method of reduction to absurdity.

Assume circle is the nearest circle in $A(P(u_i))$ to circle $C(u_{i+1})$, and there is a circle C' so that its two intersection points with circle are at the boundary of $A(P(u_{i+1}))$. Circle $C(u_j)$ intersects circle $C(u_{i+1})$ at points a and b. Circle C' intersects circle $C(u_{i+1})$ at points c and d which both at the boundary of $A(P(u_i))$. Since only both point c and point d at the arc $\overset{\frown}{ab}$ of circle $C(u_{i+1})$ ensures that points c and d are located at the boundary of $A(P(u_i))$. Because circles $C(u_j)$ and C' are nearer to circle $C(v_x)$ than circle $C(u_{i+1})$, points a and b are covered by circle C'. So circle C' is nearer to circle $C(u_{i+1})$ than circle $C(u_j)$, which contradicts the assumption. Hence there is no circle C' like this.

■

Step 5) in *algorithm* 1 requires calculating updated set $Q(v_x)$ when $A(u_{i+1})$ covers some vertices in $Q(v_x)$, which is implemented by followed sub-algorithm.

We assume that points d_k, d_{k+1}, ... , d_{k+m} in $Q(v_x)$ are covered by $A(u_{i+1})$. As points d_k, d_{k+1}, ... , d_{k+m} are inside $C(u_{i+1})$, according to *theorem* 1 the arcs among points d_k, d_{k+1}, ... , d_{k+m} must be inside circle $C(u_{i+1})$ as well. So we can let d_k adjacent to d_{k+1}, d_{k+1} adjacent to d_{k+2}, … , d_{k+m-1} adjacent to d_{k+m} by arcs.

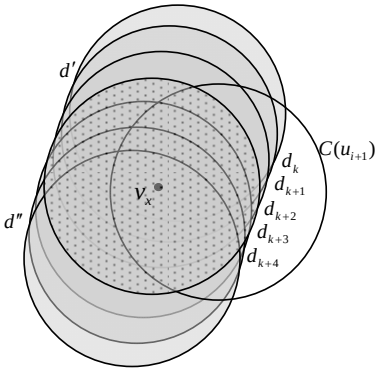


Figure 4. Added circle covers multi-intersection points of boundary

Let m=4, as shown in Fig.4 After adding circle $C(u_{i+1})$, $A(u_{i+1})$ covers points d_k , d_{k+1} , d_{k+2} , d_{k+3} , d_{k+4} in set $Q(v_x)$. If we use $b(\overset{\frown}{ab},\overset{\frown}{bc})$ to denote that point b is the intersection point of arc $\overset{\frown}{ab}$ and arc $\overset{\frown}{bc}$. If $\overset{\frown}{ab}$ is arc of circle $C(u)$ and $\overset{\frown}{bc}$ is arc of circle $C(v)$, then $b(\overset{\frown}{ab},\overset{\frown}{bc})$ can also be written as $b(u,v)$.

***Definition* 1.** b associates u and v, if b satisfies relationship $b(u,v)$.

Let arcs $\overset{\frown}{d'd_k}$, $\overset{\frown}{d_kd_{k+1}}$,, $\overset{\frown}{d_{k+2}d_{k+3}}$, $\overset{\frown}{d_{k+3}d_{k+4}}$, $\overset{\frown}{d_{k+4}d''}$ are arcs of circles respectively with u_d , u_k , u_{k+1} , u_{k+2} , u_{k+3} , u_{k+4} as the center. Points d_k , d_{k+1} , d_{k+2} , d_{k+3} , d_{k+4} can be

denoted by relationship $b(u,v)$ as: $d_k(u_d,u_k)$, $d_{k+1}(u_k,u_{k+1})$, $d_{k+2}(u_{k+1},u_{k+2})$, $d_{k+3}(u_{k+2},u_{k+3})$, $d_{k+4}(u_{k+3},u_{k+4})$. Considering that $d_k(u_d,u_k)$ and $d_{k+1}(u_k,u_{k+1})$ have a common vertex u_k, we define the new relationship of adjacent points.

***Definition* 2.** If adjacent intersection points and d_{k+1} have relationship $d_k(u_d,u_k)$ and $d_{k+1}(u_k,u_{k+1})$, thus the relationship of points d_k and d_{k+1} is defined as $\overline{d_k d_{k+1}}(u_d,u_{k+1})$, in which arc $\overline{d_k d_{k+1}}$ associates u_d, u_{k+1}.

According to the relationship among d_k, d_{k+1}, d_{k+2}, d_{k+3} and d_{k+4}, the adjacent points can be merged. For example, points d_k and d_{k+1} can be merged to be $\overline{d_k d_{k+1}}(u_d,u_{k+1})$ representing that arc $\overline{d_k d_{k+1}}$ associates u_d and u_{k+1}, then $\overline{d_k d_{k+1}}(u_d,u_{k+1})$ and $d_{k+2}(u_{k+1},u_{k+2})$ can be merged to be $\overline{d_k d_{k+2}}(u_d,u_{k+2})$. By the same method, d_k, d_{k+1}, d_{k+2}, d_{k+3} and d_{k+4} can be finally merged to be one formula $\overline{d_k d_{k+4}}(u_d,u_{k+4})$. If the whole $\overline{d_k d_{k+4}}$ is regarded as a point *b*, it can be derived that $A(u_{i+1})$ only covers one point *b*, and we get that $C(u_d)$ and $C(u_{k+4})$ are the right circles in current region whose intersection points with $C(u_{i+1})$ are at the boundary of region $A(u_{i+1})$.

Let set Q' to store covered points temporarily, set H for intermediate result. For convenience, let Q to denote $Q(v_x)$ in this algorithm.

Basing on the above analysis, the sub-algorithm to update set $Q(v_x)$ is as follows.

***Sub-algorithm*.** Update set $Q(v_x)$ in the case that boundary intersection points are covered

Input: set $Q(v_x)$, and vertex u_{i+1}.

Output: updated set $Q(v_x)$.

1) Initialize set $Q'=\varnothing$, $H=\varnothing$.

2) Pick an arbitrary unpicked point d_k from set *Q*, and calculate the distance d between d_k and u_{i+1}.

3) If $d \le R$, then put d_k into set Q'.

4) Return to step 2), until all points in *Q* are picked.

5) Let $Q = Q - Q'$.

6) Assuming the points in Q' to be $d_k, d_{k+1}, \ldots, d_{k+m}$, pick an arbitrary point d_{k+i} (i=0, 1, 2…) from Q', and put vertices u_{k+i} and v_{k+i} which associates with d_{k+i} into *H*.

7) Choose an unpicked point $d_{k+j}(j \ne i) = \{(x_{k+j}, y_{k+j}); (u_{k+j}, v_{k+j})\}$ from Q', such that $(u_{k+i}=u_{k+j}) \cup (u_{k+i}=v_{k+j}) \cup (v_{k+i}=u_{k+j}) \cup (v_{k+i}=v_{k+j})$ is true.

8) If $u_{k+i}=u_{k+j}$, then replace vertex u_{k+i} of *H* by v_{k+j}. If $u_{k+i}=v_{k+j}$, then replace vertex u_{k+i} of *H* by u_{k+j}. If $v_{k+i}=u_{k+j}$, then replace vertex v_{k+i} of *H* by v_{k+j}. If $v_{k+i}=v_{k+j}$, then replace vertex v_{k+i} of *H* by u_{k+j}.

9) Return to step 7), until there is no eligible point.

10) Let the vertices of *H* to be *u* and *v*, then calculate the points $a=\{(x_a,y_a);(u,u_{i+1})\}$ and $b=\{(x_b,y_b);(u,u_{i+1})\}$ intersected by $C(u)$ and $C(u_{i+1})$, and $c=\{(x_c,y_c);(v,u_{i+1})\}$ and $d=\{(x_d,y_d);(v,u_{i+1})\}$ by $C(v)$ and $C(u_{i+1})$.

11) If there is a point $p \in P(u_i)$ whose distance to certain one of *a*, *b*, *c*, *d* is less than *R*, then delete the certain point which may be any one of *a*, *b*, *c*, *d*.

12) Repeat step 11), until there are only two points left of *a*, *b*, *c*, *d*. Then put those two into set *Q*.

13) End.

B Solving Problem B

The forwarding set of the arbitrary vertex *v* in graph *G* can be gotten by *algorithm* 1. Before calculating CDS, we assume that *algorithm* 1 has been applied to each vertex in graph *G* and *B*(.) (. indicates arbitrary vertex in set *V*) has also been acquired already.

While calculating CDS, special packet Pa is broadcasted. Each node in the network sets its status bit ss for reducing the number of redundant dominating nodes and dominating bit cs for identifying dominating nodes. A node sets its ss to 0 if it has received packet Pa and sets its cs to 1 according to step 2) of *algorithm* 2. The steps of the *algorithm* 2 are following.

***Algorithm* 2:** Calculating CDS

Input: B(.).

Output: CDS.

1) Initialize each node's ss bit to 1 and cs bit to 0.

2) Select a sink node *s* in set *V*, and set ss bit of sink node *s* to 0.

3) Node *s* broadcasts packet Pa (including the information of $B(s)$ in head) to $N(s)$ and waits for receiving packet Pa (including status bit ss) from its neighbors. If Pa has been received, *s* sets its cs to 1. If time is out, go to step 7).

4) Each node in $N(s)$ checks its ss, and if ss equals to 0, then go to step 7).

5) Each node in $N(s)$ sets its ss to 0 and checks out if it belongs to $B(s)$. If not, then go to step 7).

6) The node in $N(s)$ takes the place of node *s*, and continue step 3), until all nodes in $N(s)$ have broadcast packet Pa already.

7) Put all nodes whoes cs equals to 1 in the connected dominating set.

8) End.

Fig.5 shows the topology of a network with 13 nodes, in which node 1 is selected to be sink node *s*. Black node indicates confirmed dominating node, and white node indicates non-dominating node. Nodes 1, 3, 4, 5, 9 and 11 are selected to be dominating nodes successively. (Considering channel delay, retransmission caused by packet loss and distance between nodes in real networks, the

connected dominating set in Fig.5 is just one of the possible results.)

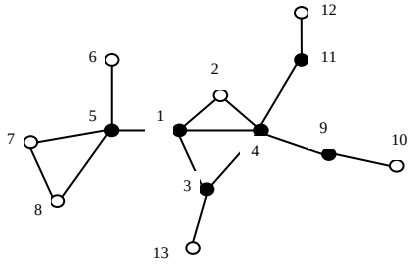


Figure 5. The connected dominating set of the network with 13 nodes

In *algorithm* 2, it effectively decreases the number of redundancy dominating nodes by setting ss. Node checks out if it will participate in transmission only by some simple calculating, and if it will, then it should set its cs to 1.

C Correctness Proof and Complexity Analysis

According to *algorithm* 2, if $G=(V,E)$ is a simple connected graph, then for arbitrary vertex v_x except the initial node, there must exist a v_y as the dominating node such that $v_x \in B(v_y)$. Hence the set (CDS) calculated by our algorithm is connected. Moreover, for an arbitrary vertex u_x in the set *V*, if u_x does not broadcast packet Pa, then there must be a vertex v_x which is confirmed to be dominating node and $u_x \in N(v_x)$ as shown in *algorithm* 1 and *algorithm* 2. Thus CDS calculated by algorithm OHCDS is the connected dominating set.

The complexity of *algorithm* 1 is mainly caused by following three parts: 1) Order the adjacent vertices of v_x by the distance to v_x. 2) Check out whether the circle $C(u_{i+1})$ added covers any boundary intersection points or not, if so, then go to *sub-algorithm*, 3) else choose a circle $C(u_j)$ nearest to circle $C(u_{i+1})$.

Suppose that Δ indicates the degree of the node. If we adopt QuickSort in the first part, the complexity of this part is $O(\Delta \log_2 \Delta)$. In the second part, while judging if the added circle covers any boundary intersection points or not, the average times needed to judge increases linearly. Thus its complexity for this part is $O(\Delta)$. As the situation that at least two boundary intersection points are covered by the added circle is dealt with in *sub-algorithm*, its time complexity is from $O(\Delta^2)$ as worst to $O(1)$ as best and $O(\Delta)$ as average. Step 11) and 12) of *sub-algorithm* are to choose vertices so that their complexity can be ignored. So the time complexity of the second task is $O(\Delta^2)$. On account of picking vertex u_j in $P(u_i)$, its comparing times increases linearly, so the time complexity of the third task is $O(\Delta)$.

Because nodes communicate with each other by broadcasting in *algorithm* 2, its time complexity can be ignored. So the time complexity of our algorithm OHCDS is $O(\Delta^2)$.

D Performance Bound Analysis

The covering problem is stated as follows: What is the minimum number of circles required to completely cover a given 2-dimensional space? It is known that no arrangement of circles could cover the plane more efficiently than the hexagons with sides *R* arrangement shown in Fig.6.

The best-case performance bound of OHCDS can be derived from covering problem. OHCDS gets approximate hexagonal vertices by calculating the minimum forwarding set and constructing a connected dominating set. It is assumed that the area of the network *A* is large compared to the area of one hexagon πR^2.The number of hexagons required to cover the entire network of area *A* is $\frac{A}{3\sqrt{3}R^2/2}$. The distance between two hexagon centers is greater than *R* and less than 2*R*, thus the best-case performance bound of OHCDS is $\frac{2A}{3\sqrt{3}R^2/2}$.

The worst-case performance bound is determined by the maximum number of transmissions. When this situation occurs, a node transmits only $R/2+\varepsilon$ distance, where ε is an infinitesimal. Thus the worst-case performance bound is $\frac{2A}{3\sqrt{3}(\frac{R}{2}+\varepsilon)^2/2}$. Ignoring ε, we can get the worst-case performance bound is $\frac{8A}{3\sqrt{3}(R)^2/2}$.

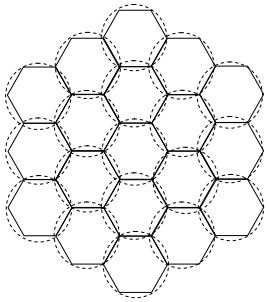


Figure 6. Covering a plane with circles in an efficient way Experimental Results

V. EXPERIMENTAL RESULTS

To test the performance of our algorithm (OHCDS), some simulation is done among our algorithm OHCDS, Dai's algorithm [13] and MISB algorithm [14] to compare their message number (communication traffic), convergence time and the size of CDS. Message number indicates how many broadcasting messages are needed by all of the nodes to accomplish the whole algorithm. Convergence time indicates how much time has been spent since the algorithm beginning to the end. Assume that time-unit means broadcasting interval between adjacent nodes and this broadcasting interval is same between any two adjacent nodes in the network. Channel delay and retransmis-

sion caused by packet loss and processing time of the node are ignored. We define convergence time as the number of broadcasting intervals since the algorithm beginning to the end. The topology of the network is formed according to the following principles:

- Nodes are randomly dispersed in a $1000m \times 1000m$ square area.
- The communication range of each node is 150m.
- The size of network is from 100 nodes to 450 nodes with increment of 50 nodes respectively.
- The simulation result is the average of 50 testing results with 8 different node groups.

In the experiments, we compare these three algorithms' performances on message number, convergence time and the size of CDS based on different node density and show results in Fig.7, 8, and 9. Experiments are as following.

Experiment 1 Compare the message number of our algorithm with those of Dai's algorithm and MISB. The result is as Fig.7 shows.

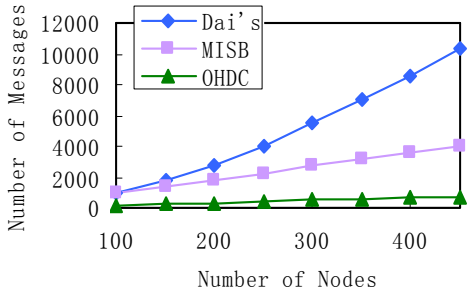


Figure 7. Comparison of the message number among OHCDS algorithm, Dai's algorithm and MISB algorithm

As OHCDS uses only 1-hop adjacent node information, the message number of OHCDS is mainly in the process of neighbor discovery and broadcasting in *algorithm* 2. The message number of neighbor discovery equals to the number of nodes, and the message number of broadcasting in *algorithm* 2 equals to the number of nodes that broadcast messages including the minimum forwarding set information. For this reason, the message number of our OHCDS algorithm is less than the double number of nodes in the network. As Fig.7 shows, the message number of Dai's algorithm is about 20 times the number of nodes, while the message number of MISB algorithm is about 9 times the number of nodes. So OHCDS is much better.

Experiment 2 Compare the convergence time of our algorithm with those of Dai's and MISB. To simplify the simulation, we assume that the neighbor discovery time in our algorithm is 1. The result is as Fig.8 shows.

Because *algorithm* 1 is executed severally, most of time is consumed by *algorithm* 2. Furthermore, it adopts broadcasting diffusion in *algorithm* 2, so time spent would not be more than the maximum route length. However, time spent on MISB algorithm is about 5 times the route length. Fig.8 depicts the comparison of the convergence time among these three algorithms with different node density. We can see that the convergence time of OHCDS approximately equals to that of Dai's algorithm with very low node density. However, with node number increasing in the network and node density increasing, the convergence time of Dai's algorithm increases rapidly, but the convergence time of OHCDS is almost invariant as the region is constant. We can also see in Fig.8 that the convergence time of OHCDS is one fifth of that of MISB.

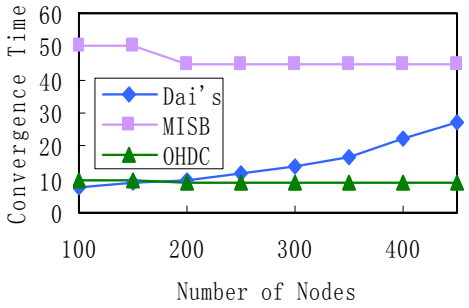


Figure 8. Comparison of the convergence time among OHCDS algorithm, Dai's algorithm and MISB algorithm

Experiment 3 Compare the CDS size of our algorithm with those of Dai's and MISB.

As shown in Fig.9, the CDS size of our algorithm is between that of Dai's algorithm and MISB algorithm. With node density increasing, the size of CDS of our algorithm is increasing obviously. It is mainly because of using communication coverage method in algorithm 1.

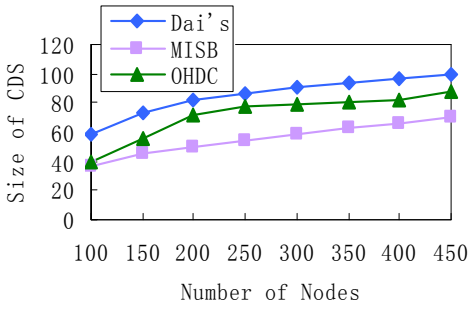


Figure 9. Comparison of the CDS size among OHCDS algorithm, Dai's algorithm and MISB algorithm

Summing up the above, we can see that our algorithm has the best performance on both message number and convergence time and also get small CDS size.

VI. Conclusion

Comparing to the other algorithms, there are a lot of advantages of our algorithm due to calculating connected dominating set using only 1-hop neighbor information. As simulation results show, less communication cost and less convergence time is achieved by our algorithm. Less message number means less energy consumption, and that less convergence time indicates faster calculating. Our algorithm can also get a smaller CDS size with lower node density. Meanwhile, our algorithm is low energy consumption and low convergence time, and it is applicable to dynamic wireless sensor networks. As for static network, our algorithm achieves energy balance. The future work will focus on optimizing the CDS size on the assumption of keeping its present advantages.

ACKNOWLEDGMENT

This paper is supported by Research Fund for the Doctoral Program of Higher Education of China for New Teachers No.200806141110 and project CNGI of NDR No.CNGI-09-01-07.

REFERENCES

[1] S.C.H. Huang, M. Sun, P. Wan, X. Jia, "Interference-Aware, Fully-Distributed Virtual Backbone Construction and its Application in Multi-Hop Wireless Networks", IEEE Transactions on Communications, vol.58, no.12, pp. 3550-3560, 2010.

[2] Dai F, Wu J, "Performance analysis of broadcast protocols in adhoc networks based on self-pruning", IEEE Trans on Parallel and Distributed Systems, vol. 15,no.11, pp. 1-13, 2004.

[3] Lee J, Mans B, "Energy-efficient virtual backbones for reception-aware MANET", Proceedings of the 63rd IEEE Vehicular Technology Conference, Melbourne: IEEE Press, 2006: 1097-1101.

[4] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and Ker-I Ko, "A greedy approximation for minimum connected dominating sets", Theoretical Computer Science, vol. 329, issues1–3, pp. 325–330, Dec. 2004.

[5] S. Butenko, X. Cheng, C. Oliveira, P. M. Pardalos, "A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks", Recent Developments in Cooperative Control and Optimization, pp. 61-73, New York:Kluwer Academic Publishers, 2004.

[6] M. Min, H. Du, X. Jia, C. Huang, S. Huang, and W. Wu, "Improving construction for connected dominating set with Steiner tree in wireless sensor networks", Journal of Global Optimization, vol.35, no.1, pp.111-119, 2006.

[7] Lu G, Zhou MT, Tang Y, Zhao MY, Niu XZ, and She K, "Approximation algorithms for the connected dominating set problem in unit disk graphs", Journal of Electronic Science and Technology of China, vol.3, no.7, pp. 214-222, 2009.

[8] A. Vahdatpour, F. Dabiri, M. Moazeni, and M. Sarrafzadeh, "Theoretical Bound and Practical Analysis of Connected Dominating Set in Ad Hoc and Sensor Networks", Distributed Computing, vol.5218, pp.481-495, 2008.

[9] Joseph P. Macker, "Distributed connected dominating set election from uniform random to power law network graphs", Military Communications Conference, pp.1-7, 2009.

[10] Wu J, Lou W, Dai F, "Extended multipoint relays to determine connected dominating sets in MANETs", IEEE Transactions on Computers, vol.55, no.3, pp.334-347, 2006.

[11] H. Du, Q. Ye, J. Zhong, Y. Wang, W. Lee and H. Park, "PTAS for Minimum Connected Dominating Set with Routing Cost Constraint in Wireless Sensor Networks", Combinatorial Optimization and Applications, vol.6508, pp.252-259, 2010.

[12] Dai F, Wu J, "An extended localized algorithm for connected dominating set formation in ad hoc wireless networks", IEEE Transactions on Parallel and Distributed Systems, vol.15, no.10, pp.905-920, 2004.

[13] Tang Y, Zhou M, "Maximal Independent Set Based Distributed Algorithm for Minimum Connected Dominating Set", Acta Electronica Sinica, vol.35, no.5, pp.868-874, 2007.(in Chinese)

[14] Sakai K., Sun M, Ku W. "Fast Connected Dominating Set Construction in Mobile Ad Hoc Networks", IEEE International Conference on Communications (ICC'09), Germany: IEEE Press, 2009: 1-6.

[15] Wenyong Wang, Jun Zhang, Yong Tang, Yu Xiang, Ting Yang, "One-Hop Neighbor Transmission Coverage Information Based Distributed Algorithm for Connected Dominating Set", 3rd International Symposium on Parallel Architectures, Algorithms and Programming, pp.15-21, December 2010.

Wang Wenyong was born in Sichuan, China, in 1967. He is a professor of University of Electronic Science and Technology of China. He received his Ph.D. from University of Electronic Science of China. He is a member of China Education and Research Network Committee of Experts, senior member of China Computer Federation, member of Internet Professional Committee, member of Software Engineering Professional Committee, and member of High Performance Computing Professional Committee. His research interest lies in wireless sensor network and network computing.

Zhang Jun was born in Gansu, China, in 1985. He is a engineer of Huawei Technologies Company. He received his M.S. from University of Electronic Science and Technology of China in 2011.

Tang Yong was born in Yunnan, China, in 1973. He is an associate professor of University of Electronic Science and Technology of China. He received his Ph.D. from University of Electronic Science and Technology of China in 2007, M.S. from Xidian University in 1999, and B.S. from Xidian University in 1996. Prior to reading his doctorate in computer science, he served as a senior engineer for DSET (China), working on the projects of TMN. His research interest lies in wireless sensor network and network management.

Xiang Yu was born in Sichuan, China, in 1973. He is an associate professor of University of Electronic Science and Technology of China. He received his B.S., M.S., Ph.D. from University of Electronic Science of China. His research focuses upon wireless sensor network and wireless communication.

Yang Ting was born in Sichuan, China, in 1975. He is a lecturer of University of Electronic Science and Technology of China. He received his M.S. from University of Electronic Science of China. He is currently a Ph.D. candidate in Computer Science in University of Electronic Science and Technology of China. His research fields include wireless sensor network and network security.

An Internet Traffic Identification Approach Based on GA and PSO-SVM

Jun Tan
School of Computer Science, Sichuan University, Chengdu, China
Email: chinatanjun@gmail.com

Xingshu Chen and Min Du
School of Computer Science, Sichuan University, Chengdu, China
Email: chenxsh@scu.edu.cn, doingscu@gmail.com

***Abstract*—Internet traffic identification is currently an important challenge for network management. Many approaches have been proposed to classify different categories of Internet traffic. However, traditional approaches only focus on identifying TCP flows and have ignored the selection of best feature subset for classification. In this paper, we propose an approach to classify both TCP and UDP traffic flows using the Support Vector Machine (SVM) algorithm. In this approach, we select the best feature subset using Genetic Algorithm, and then we calculate the correspondence weight of each feature selected by Particle Swarm Optimization (PSO). In addition, the traditional SVM algorithm is optimized by PSO algorithm. The experimental results demonstrate that this approach can effectively select the feature subset from multiple attributes that can best reflect the differences among different network applications. Moreover, the identification rate is improved by the method of feature weighting and PSO optimized SVM algorithm.**

***Index Terms*—Traffic Identification, Genetic Algorithm, Particle Swarm Optimization, Support Vector Machine, Statistical Characteristics**

I. INTRODUCTION

With the rapid development of Internet technology, an increasing number of network applications have emerged. However, these applications may result in serious network and security problems. For example, some network applications may take up large bandwidth, congest network access, and reduce network performance greatly [1]. Also, some applications may leak users' privacy [2]. To solve these problems, it is very necessary to identify and control Internet traffic flows effectively, which is of great significance to network management, traffic control and so on.

Three major approaches have been proposed to identify Internet traffic flows: port based approach [3], Deep Packet Inspection (DPI) approach [4] and the statistical-based approach based on statistical characteristics of flows' behavior [5-7]. However, these applications could have potential defects. For example, current network applications may use dynamic ports to communicate with each other, and network flows among these applications cannot be easily detected by port based approach. Some applications may encrypt their data during the communication. Therefore, the DPI approach cannot effectively identify the traffic flows for these applications.

Recently, traffic identification approaches using statistical characteristics have attracted a lot of attention, and many algorithms such as machine learning and neural network have been used to classify different categories of traffic flows. However, there are still some areas not considered before.

First of all, feature selection is one of the most critical steps in the problem of Internet traffic identification. We can classify different categories of applications is because there exist some discrepancies of their behaviors. However, researchers at present choose one or more features from a variety of characteristics to classify different traffic flows only based on the qualitative analysis of different features. Different researchers have different opinions on the importance of features of traffic flows.

Secondly, although a few features have been chosen to classify different Internet traffic, not every feature has the same importance. Therefore, in order to improve the recognition rate, each feature selected could have a weight value representing its importance.

Thirdly, previous works only focused on identifying TCP flows, and traffic flows using UDP protocol cannot be identified.

Therefore, in this paper, we propose an approach to identify both TCP and UDP traffic flows using Support Vector Machine (SVM). The main contributions are described as follows:

(1) In this approach we classify Internet traffic based on the statistical characteristics of traffic flows without

Manuscript received December 27, 2010; accepted March 21, 2011.
Project Number: 973 Key Basic Research Program (No.2007CB311106), corresponding author: Xingshu Chen.

doi:10.4304/jcp.7.1.19-29

using any port or host information, and there is no need to inspect the signature of applications for privacy consideration.

(2) The concepts of TCP session and UDP session are defined precisely in this approach, making the proposed algorithm can classify both TCP and UDP traffic flows effectively.

(3) Quantitative methods have been used to choose the feature subset from multiple features by Genetic Algorithm (GA) that can best distinguish different applications, and a corresponding weight is assigned to each feature by Particle Swarm Optimization (PSO) algorithm.

(4) SVM algorithm is used to classify different traffic flows, and we optimize the traditional SVM algorithm by PSO, which can improve the performance of SVM algorithm effectively.

(5) The experimental results show that this approach can classify different Internet traffic at a high rate. A classification accuracy of 97% in the situation of training on one day of traffic from one site, and testing on traffic from that site for a day eight months later, which verifies the robustness and practicability of this algorithm.

The rest of the paper is organized as follows. Section Ⅱ briefly describes some related work in this field; Section Ⅲ describes the basic concepts and the system overview; Section Ⅳ describes the feature subset selection algorithm based on GA; Section Ⅴ describes the calculation process of features' weights and the PSO optimized SVM algorithm. Section Ⅵ describes the process of data collection, and then presents and discusses the experimental results. Section Ⅶ presents the conclusion and future work.

II. RELATED WORK

The identification of network applications through traffic flows is important to some certain fields, such as traffic control, quality of service and so on. Traditionally, network applications use default port to communicate with each other, and ISPs can effectively identify and classify network traffic [3]. In order to hide the traffic, some network applications have started using dynamic port for communication; some applications even use the TCP port 80 to communicate.

To solve the above problems, a new method which rely on application payload signatures were developed, which is also called Deep Packet Inspection (DPI) [4]. This approach directly compares the stored signatures to the packets from applications to accurately classify them. This method has the advantages of accuracy and real-time [8]. In addition to signature-based methods, other payload-based methods are also proposed. ACAS uses the first N bytes of payload as input to train a machine learning model to classify flows [9]. However, as some applications' protocols keep upgrading and new network applications emerge, also some applications even encrypt their data, making the DPI method not useful any more.

Given the shortcomings of port and signature based approaches for detecting Internet traffic, many researchers classify network traffic flows based on the statistics of features of different traffic flows. Karagiannis first proposed the identification method based on statistical characterization [10], and after that there were a few methods based on the statistics of traffic flows [11-14]. These methods collect and make statistics of traffic flows, such as the distribution of packet size in a flow, the duration of a flow, the interval of arrival time of each packet. Sebastian first introduced the machine learning method to the traffic recognition, and used the unsupervised Bayesian learning algorithm for traffic classification [15]. Then a lot of network traffic identification algorithms appeared using the algorithms in machine learning, such as Support Vector Machine [16-17], decision tree algorithm [18] and neural network algorithm [19-20].

The identification methods based on the statistical characteristics of traffic flows have played an significant role in classifying different categories of network applications, however, the previous works mainly focused on the identification of TCP traffic flows, and UDP traffic flows were not considered [19,21]. Moreover, little work has been done on the selection of attributes of traffic flows.

III. PRELIMINARIES

A. Basic Concepts

A packet transmitted in the network is the manifestation of a network application. However, a single packet carries too little information and has little effect on the performance of traffic identification. Therefore, a single packet should be combined with other relevant packets to form a session, and traffic flows can be classified by the form of a session.

Definition 1 (TCP Session Flow): If host A and host B communicate with each other through TCP protocol, and their corresponding communication ports are *portA* and *portB*. Suppose p is a TCP session flow, and then p includes all the data packets transmitted between *portA* and *portB*. In particular, these data packets transmission are initialized after a SYN packet has been sent, and are ended before the first FIN packet will be sent.

Definition 2 (UDP Session Flow): If host C and host D communicate with each other through UDP protocol, and their corresponding communication ports are *portC* and *portD*. Suppose p is a UDP session flow, and then p includes all the data packets transmitted between *portC* and *portD* within time *t*.

TCP session flow and UDP session flow are collectively referred to as session flow in this paper.

Definition 3 (Feature Vector of Network Traffic): Suppose $s(p)$ is an attribute of session p, then $\boldsymbol{R}^D(p) = [s_1(p), s_2(p), \cdots, s_D(p)]^{\mathrm{T}}$ is the feature vector of D attributes of session p. $\boldsymbol{R}^D(p)$ is called the feature

vector of network traffic.

There were 246 per-flow features used for classification in [19]. Most of these features can only be used for classifying TCP traffic flows, while having no use for UDP protocol. However, many more network applications are using UDP protocol for transmission nowadays, such as P2P applications. So it is also important to classify UDP traffic flows. In this approach, we analyzed the characteristics of many Internet traffic session flows and selected 45 features to identify different network applications using both TCP and UDP protocols. All the features of a session are shown in Appendix A.

Definition 4 (Network Identification Model): Network traffic identification model is a 6-tuple of $(Q, \boldsymbol{R}^D, \boldsymbol{R}^d, \delta, \varphi, q)$, and it subjects to the following requirements:

1) Q is a finite set, $Q = \{a \mid a \, is \, network \, protocol\}$

2) $\boldsymbol{R}^D$ is feature space of D-dimension, representing the original feature vector

3) $\boldsymbol{R}^d$ is feature space of d-dimension, representing the feature vector after the feature selection process

4) $\delta : \delta(\boldsymbol{R}^D) \to \boldsymbol{R}^d$ is the selection operator, choosing the most important features from the original feature vector

5) $\varphi(\boldsymbol{R}^d) \to q$, φ is classification operator, q is the network protocol of feature vector $\boldsymbol{R}^d$ after classifying, and $q \in Q$.

B. Traffic Categories

Fundamental to classification work is the idea of classes of traffic. In the present data set, traffic was classified into common groups according to the way how they transmitted their packets. The traffic was classified into 10 categories in [19], however, as P2P technology has been widely used in recent years. Therefore, we classify Internet traffic into 12 categories as listed in Table 1.

TABLE I.
NETWORK TRAFFIC CATEGORIES

Classification	Example Applications
WWW	Web
P2P (File Sharing)	eDonkey, BitTorrent
P2P (Multimedia)	Sopcast, PPStream, PPTV
P2P (Instant messaging)	Skype, MSN
ATTACK	Worm, Virus
MULTIMEDIA	Windows Media Player
GAMES	WOW
MAIL	IMAP, POP, SMTP
INTERACTIVE	ssh, klogin, rlogin
DATABASE	postgres, sqlnet
BULK	ftp
SERVICES	X11, DNS, LDAP

C. Particle Swarm Optimization (PSO)

PSO is an evolutionary computation technique developed by Kennedy and Eberhart in 1995, which searches for the best solution by simulating the movement and flocking of birds [22]. PSO is an optimization tool, providing a population-based search procedure in which individuals called particles change their positions with time. In a PSO system, the group is a community composed of all particles, and all particles fly around in a multi-dimensional search space. Each particle flies with a certain velocity and finds the global best position after some iteration. At every iteration period, each particle adjusts its velocity vector based on its momentum and the influence of its best position in history as well as the best positions of all particles, then a new position is computed by the velocity and its position in the last iteration.

Suppose the dimension for a searching space is n, the total number of particles is m, and the population is presented as $X = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_i, \cdots, \boldsymbol{x}_m)^T$, the position of the ith particle is presented as vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T$. The velocity of the ith particle is presented as vector $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \cdots, v_{in})^T$, and the best position of this particle being searched until now is denoted as $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \cdots, p_{in})^T$, the best position of the whole population being searched until now is denoted as $\boldsymbol{p}_g = (p_{g1}, p_{g2}, \cdots, p_{gn})^T$. The rule of adjusting the velocity and position of a particle is described as:

$$v_{id}^{(t+1)} = w^{(t)} v_{id}^{(t)} + c_1 r_1 \left(p_{id}^{(t)} - x_{id}^{(t)} \right) + c_2 r_2 \left(p_{gd}^{(t)} - x_{id}^{(t)} \right) \quad (1)$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \quad (2)$$

where $i = 1, 2, \cdots m$ represents the number of particles, $d = 1, 2, \cdots n$ represents the searching space of this problem, t is the current evolution generation, r_1 and r_2 are random numbers between 0 and 1, c_1 and c_2 are the acceleration constants with positive values, w is the inertia weight [23-24]. The general steps involved in PSO are illustrated in Algorithm 1, in which the function $f()$ is the fitness function.

D. Support Vector Machine (SVM)

The purpose of SVM classification is to find optimal separating hyper-plane by maximizing the margin between the separating hyper-plane and the data. Give a set of data $T = \{\boldsymbol{x}_i, y_i\}_{i=1}^m$, where $\boldsymbol{x}_i$ denotes the input vectors, $y_i \in \{+1, -1\}$ stands for two classes, and m represents the number of training data [25-26].

SVM attempts to identify a hyper-plane, which functions as a separating plane for classification of data in

a multidimensional space. The hyper-plane $f(\boldsymbol{x})$ is presented as:

$$f(\boldsymbol{x}_i) = w \bullet \boldsymbol{x}_i + b = 0, i = 1, 2, \cdots m \tag{3}$$

Algorithm 1 Particle Swarm Optimization

```
Initialize population
for gen = 1 : maxgen do
  for i = 1 : m do
    if f( x_i ) < f( p_i ) then
      p_i ← x_i
    end if
    if f( p_i ) < f( p_g ) then
      p_g ← p_i
    end if
  end
  update the velocity and position of each particle
according to (1) (2)
  if minimum error criteria is attained then
    break;
  end if
end
```

where w denotes the weight vector, and b denotes the bias term. w and b are used to define the position of separating hyper-plane.

The separating hyper-plane should satisfy the following constraints:

$$y_i f(\boldsymbol{x}_i) = y_i \left(w \bullet \boldsymbol{x}_i + b \right) \geq 1, i = 1, 2, \cdots m \tag{4}$$

Positive slack variables ξ_i are introduced to measure the distance between the margin and the vectors $\boldsymbol{x}_i$ that lying on the wrong side of the margin. Then, the optimal hyper-plane separating the data can be obtained by the following optimization problem:

$$\begin{aligned} &\text{Min} \ \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i, i = 1, 2, \cdots m \\ &\text{s.t.} \quad y_i \left(w \bullet x_i + b \right) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0 \end{aligned} \tag{5}$$

where C is the error penalty.

By the Lagrangian multiplier, $a_i, i = 1, 2, \cdots m$ is introduced, the previous mentioned optimization problem is transformed into the dual quadratic optimization problem as below:

$$\begin{aligned} &\text{Max} \ \ L(a) = \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i,j=1}^{m} a_i a_j y_i y_j x_i x_j \\ &\text{s.t.} \quad \sum_{i=1}^{m} a_i y_i = 0, a_i \geq 0, i = 1, 2, \cdots m \end{aligned} \tag{6}$$

Therefore, the linear decision function is created by solving the dual optimization problem, which is defined as:

$$f(x) = \text{sign}\left(\sum_{i,j=1}^{m} a_i y_i \left(\boldsymbol{x}_i, \boldsymbol{x}_j \right) + b \right) \tag{7}$$

In most cases, the data are not linearly separable, and are consequently mapped to a higher-dimensional feature space. Therefore, if the data cannot be classified clearly in the current dimensional space, then the SVM will map them to a higher dimensional space for classification. In order to construct the feature space, the non-linear mapping function $\phi(x)$ is used. Then, the non-linear decision function is expressed as follows:

$$f(x) = \text{sign}\left(\sum_{i,j=1}^{m} a_i y_i K\left(\boldsymbol{x}_i, \boldsymbol{x}_j \right) \right) + b \tag{8}$$

$$K\left(x_i, x_j \right) = \phi\left(x_i \right) \phi\left(x_j \right) \tag{9}$$

where $K\left(x_i, x_j \right)$ is called the kernel function. Several kernel functions help the SVM in obtaining the optimal solution. The most frequently used kernel functions are the polynomial, sigmoid and radial basis kernel function (RBF). The RBF is generally applied most frequently, because it can classify multi-dimensional data. Additionally, the RBF has fewer parameters to set than a polynomial kernel. Therefore, in our approach, the RBF kernel function is used in the SVM to obtain optimal solution. It is presented as:

$$K\left(x_i, x_j \right) = \exp\left(-\gamma \left\| x_i - y_i \right\|^2 \right) \tag{10}$$

Here, C and γ should be set appropriately. Parameter C represents the cost of the penalty. The choice of the value for C influences the classification outcome. if C is too large, the classification accuracy rate is very high in the training phase, but very low in the testing phase. If C is too small, the classification accuracy cannot be satisfied. Parameter γ is more important, because its value affects the partitioning outcome in the feature space. An excessively large value for parameter γ results in over-fitting, and a disproportionately small value will lead to under-fitting. In this approach, we use PSO algorithm to optimize these two parameters.

E. Approach Overview

The training and testing processes of the identification system are presented as follows:

(1) Capture all the packets and reorganize the packets into traffic session flows according to definition 1 and definition 2, then divide the traffic flows into training set and test set.

(2) Calculate the features' statistics of traffic flows according to Appendix A.

(3) Feature selection by GA.

(4) Feature weighting and SVM algorithm optimized by PSO using the data in the training set.

As both TCP and UDP flows are classified by this algorithm, but there are biggish distinctions between TCP and UDP protocols in the form of transmission. So in this approach, two SVM models are used for training and classifying TCP and UDP flows. One is for TCP flows and the other is for UDP. There are two SVM models after the training process, and for a traffic session flow to be classified, first check if it is TCP or UDP protocol, and then classify it using the corresponding SVM model.

The system overview is shown in Figure 1.

Ⅳ. FEATURE SELECTION PROCESS

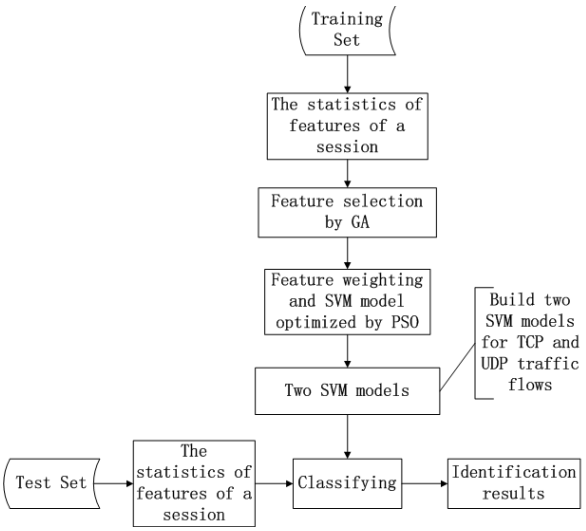


Figure 1. System Overview

A. Separable Criterion Based on Class Distance

Definition 5 (Selection of Traffic Feature): Suppose $\boldsymbol{R}^D = [r_1, r_2, \cdots, r_D]^{\mathrm{T}}$ is the feature vector of a traffic flow, selection of traffic features means choosing a subset $\boldsymbol{R}^d$ in vector $\boldsymbol{R}^D$ such that $\boldsymbol{R}^d = [r_{i1}, r_{i2}, \cdots, r_{id}]^{\mathrm{T}}, d \le D$, and features selected can make the separable criterion of different traffic flows reach the maximum value.

Separable criterion function is used to classify different traffic flows, and in the pattern recognition field, different classes could be separated is because every class is in specific region of the feature space. Moreover, the smaller the distance of samples in the same class and the larger the distance of samples of different classes are, the better the classification result is. This scheme is called the separable criterion.

Set $J = \dfrac{d_B}{d_\omega}$ as the separable criterion function, and d_B represents the sum of distance of samples of different classes, d_ω represents the sum of distance of samples in the same class. If d_B increases and d_ω decreases, the separable criterion function J gets larger, making different classes classified much easier.

In a D-dimensional feature space, set $\boldsymbol{r} \in \boldsymbol{R}^D$, and $\delta(\boldsymbol{r}_i, \boldsymbol{r}_j)$ represents the distance between two samples in feature space. Suppose there are N samples $\{\boldsymbol{r}_l, l = 1, 2, \cdots, N\}$ belonging to c classes, and the pattern of class i is defined as follows:

$$\omega_i = \{\boldsymbol{r}_k^{(i)}, k = 1, 2, \cdots, N_i\}, i = 1, 2, \cdots c \tag{11}$$

Defines:

$\boldsymbol{m}^{(i)}$: The mean vector of pattern of samples in class i

$\boldsymbol{m}$: The mean vector of pattern of all samples.

Then:

$$\boldsymbol{m}^{(i)} = \frac{1}{N_i}\sum_{k=1}^{N_i} \boldsymbol{r}_k^{(i)}, i = 1, 2, \cdots, c \tag{12}$$

$$\boldsymbol{m} = \frac{1}{N}\sum_{l=1}^{N} \boldsymbol{r}_l \tag{13}$$

The sum of pattern distance of the same class d_ω and the sum of distance of different classes d_B are defined respectively as:

$$d(\omega_i) = \frac{1}{N_i}\sum_{k=1}^{N_i} \delta(\boldsymbol{r}_k^{(i)}, \boldsymbol{m}^{(i)}), i = 1, 2, \cdots, c \tag{14}$$

$$d_\omega = \sum_{i=1}^{c} P_i d(\omega_i) = \sum_{i=1}^{c} P_i \frac{1}{N_i}\sum_{k=1}^{N_i} \delta(\boldsymbol{r}_k^{(i)}, \boldsymbol{m}^{(i)}), P_i = \frac{N_i}{N} \tag{15}$$

$$d_B = \sum_{i=1}^{c} P_i \delta(\boldsymbol{m}^{(i)}, \boldsymbol{m}), P_i = \frac{N_i}{N} \tag{16}$$

Then the separable criterion function J is presented as:

$$J = \frac{d_B}{d_\omega} = \frac{\sum_{i=1}^{c} P_i \delta(\boldsymbol{m}^{(i)}, \boldsymbol{m})}{\sum_{i=1}^{c} P_i \frac{1}{N_i}\sum_{k=1}^{N_i} \delta(\boldsymbol{r}_k^{(i)}, \boldsymbol{m}^{(i)})} \tag{17}$$

B. Feature Selection by Genetic Algorithm

The GA is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and searching problems [27]. In the problem of network traffic identification, as there are many features, it is necessary to select the optimal subset which can best classify different protocols, therefore, we choose GA here as a selection operator.

(1) Fitness function

Every individual in the population corresponds to a solution in the optimization problem, in this approach, in order to classify different traffic flows better, the value of separable criterion function should be as large as possible, therefore, we use the function J mentioned before as the fitness function:

$$Fitness = J = \frac{d_B}{d_\omega} = \frac{\sum_{i=1}^{c} P_i \delta(\boldsymbol{m}^{(i)}, \boldsymbol{m})}{\sum_{i=1}^{c} P_i \frac{1}{N_i}\sum_{k=1}^{N_i} \delta(\boldsymbol{r}_k^{(i)}, \boldsymbol{m}^{(i)})} \tag{18}$$

Euclidean distance is used here as the measurement of distance between two samples.

(2) Encoding and decoding

In GA, every solution to the problem is encoded to a chromosome, and a feature may be selected or not in the feature selection process. In order to select a combination of d features from the original D-dimensional features. A binary string of D bits is used to describe the feature combination. Digital 1 means this feature is selected, while digital 0 means not. At last, the bits with value 1 in

Feature 1	Feature 2	Feature 3	…	…	Feature D

…

1	0	1	…	…	1

Figure 2. Encoding Method

the binary string indicate the features selected as the input of classification process. The binary string is shown in figure 2.

(3) Initial population

Set N as the number of individuals in population, then the initial population is defined as $P_0 = \{\boldsymbol{x}_i\}, i = 1, 2, \cdots, N$, every $\boldsymbol{x}_i$ represents a solution, and it is a binary string of D bits as defined before. While generating the initial population, set d bits to 1, and these d features make the fitness function the largest d values, and then set the other D-d bits to 0. This algorithm is presented in algorithm 2 as below.

Algorithm 2 Generating Initial Population

Inputs: training data, D: feature space, d: feature subset space

Outputs: the initial population InitPop with d features set

InitPop = zeros(1, D); /* Create the D-bit binary string representing the features to be selected, and all bits are set to 0 */

for i = 1 : D **do**

fit[i] = the fitness value of ith feature

end

index = findmax(fit, d); /* index represents the array index of the largest d elements in array fit */

InitPop[index] = 1; /* Set 1 to the corresponding features */

return InitPop

4) Genetic operations

We use tournament selection method as the selection operator and the two-point crossover operation as the recombination of chromosomes. Its specific operation is as follows: First set two cross points randomly at the chromosome of parent population, then exchange the chromosome at the two cross points of two individuals. In order to improve the locality search ability of GA to avoid prematurity, mutation operator is imported in the genetic process. In this approach, as the number of 1 and 0 representing the features doesn't change, inversion operator is used as the mutation operator, it operates as follows: a new binary string is generated by choosing randomly two gene positions and inversing the substring between the two positions in a chromosome. The network traffic feature selection algorithm based on GA is presented in algorithm 3.

5. Feature weighting by PSO-SVM

In this approach, we use the PSO algorithm to optimize the parameters C and γ of SVM as well as the weights of the features selected by GA in the previous process.

In the previous section, a subset of features are selected for classification by GA, however, not all the features selected have the same importance. In order to increase the identification rate, each feature selected is assigned a weight value w_i, which represents the ith feature's importance. In this approach, we set the limit of a weight's value as $w_i \in [0, 2]$,.

Algorithm 3 Feature Selection by GA

Inputs: training data, D: feature space, d: feature subset space

Outputs: feature subset of d-dimensional

Population initialization: population size N, probability of crossover P_c, probability of mutation P_m, maximal evolution generations GEN

t←0

Generate the initial population $P(t)$ using algorithm 2, and calculate the fitness value of each individual

while t≤GEN **do**

use tournament selection operator to select the largest N individuals of $P(t)$ by their fitness value

according to mutation probability P_c and then use mutation operator to produce N intermediate individuals from the previous N individuals selected

calculate the fitness value of the N intermediate individuals, and choose the smallest $N \times P_m$ individuals, modify these individuals using the mutation operator

compete the N intermediate individuals against the original individuals in population $P(t)$, and select the largest N individuals according to their fitness values as the individuals of population $P(t+1)$

t←t+1

end while

Find the largest individual by fitness value from $P(t)$, the bits with value 1 of the binary string indicate the features selected as the measurement of classification

Definition 6 (Weighted Feature Vector): Suppose a d-dimensional feature space $\boldsymbol{R}^d = [r_1, r_2, \cdots, r_d]^{\mathrm{T}}$, the weighted feature vector $\boldsymbol{WR}^d$ is defined as:

$$\boldsymbol{WR}^d = [w_1 \times r_1, w_2 \times r_2, \cdots, w_d \times r_d]^{\mathrm{T}} \tag{19}$$

In this approach, we use SVM to classify different categories of Internet traffic, therefore, two decision variables, designated C and γ are required. For the feature weighting process, if d weights will be calculated, then 2+d decision variables must be adopted. Therefore, the encoding strategy is presented as follows:

$$Particle = (C, \gamma, w_1, w_2, \cdots, w_d) \tag{20}$$

To get the highest recognition rate, every feature's weight w_i will be calculated. The training data is used for training the SVM model, and the fitness function is defined as:

$N_{correct}$: The number of traffic flows identified correctly

N_{total} : The number of all traffic flows

$$Fitness = \frac{N_{correct}}{N_{total}} \quad (21)$$

SVM is used as the classification method here and the distance of different classes is measured according to the weighted feature vector as defined in definition 6.

The feature weighting calculating algorithm is described in algorithm 4:

Algorithm 4 Feature Weighting Based on PSO

Inputs: training data, *d*-dimensional features

Outputs: weights of *d* features, C and γ of SVM

Initial operation, set the current iteration t=0, and then set the maximal iteration GEN, c_1, c_2

Generate m particles $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_m$ in the (2+d)-dimensional searching space, they constitute the initial population $W(t)$, and $\boldsymbol{w}_i, i = 1, 2, \cdots m$ is the solution in the (2+d)-dimensional space

Generate the initial velocity of every particle as $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m$, set the global best solution p_g and the individual best solution p_i of m particles to 0

Every particle in population $W(t)$ is a weight of a feature, parameters C, γ. SVM is used here to train the data in training set

while t≤GEN **do**

for i = 1 : m **do**

compare the fitness value of particle i $p_{current}$ with its best solution p_i so far, if $p_{current} > p_i$, then $p_i = p_{current}$, and record the value of current particle $\boldsymbol{w}_i$

compare the fitness value of particle i $p_{current}$ with the global best solution p_g, if $p_{current} > p_g$, then $p_g = p_{current}$ and record the value of current particle $\boldsymbol{w}_i$

end

update the velocity and position of every particle in population $W(t)$ and get new population $W(t+1)$, compute the new fitness of every particle

t←t+1

end while

Output the best global solution p_g, which is the largest recognition rate with d features selected

return the value of the particle with fitness value p_g as the weights of d features and parameters C, γ

VI. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experiment Design

The data collection environment is presented in figure 3.

In a data collection process, all the testing machines

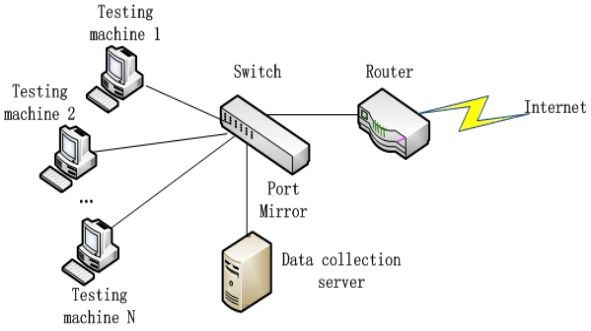


Figure 3. Data Collection Environment

are running network applications belonging to one specific category, so all the traffic flows collected belong to only one class, and these flows are collected by the data collection server through port mirroring method. Total up to 40000 flows are collected for a specific traffic category. Following these steps, all the traffic flows belonging to the 12 classes listed in Table 1 are collected, so that one data set contains 12*40000 traffic flows.

In order to verify the effectiveness of the method proposed in this approach, two separate data sets are collected. The first data set was collected from February 8th 2010 to February 14th 2010, and the second data set was collected from October 8th 2010 to October 14th 2010. These two data sets are separated by eight months, and are denoted as set 1 and set 2 respectively.

The experimental process is described as follows: 50% of the data is chosen randomly from set 1 as the training data, and the data in set 2 is used for testing. This experiment is repeated ten times; using different randomly chosen training sets of the same size, the mean and standard deviation across the repeated experiments are calculated. For some applications in table 1, they use both TCP and UDP packets to transmit, such as P2P file sharing applications. The identification rates of these classes are computed as follows:

Suppose the proportion of TCP flows in one specific class is tcp_p, the proportion of UDP flows is udp_p, they can be computed from the 40000 flows collected. And the identification rate of TCP and UDP flows are tcp_r and udp_r respectively, therefore, the identification rate for this class is presented as:

$$rate = tcp\_p * tcp\_r + udp\_p * udp\_r \quad (22)$$

where $tcp\_p + udp\_p = 1$.

B. Determination of Interval Time in UDP Session

As described in definition 2, the UDP session flow includes all the packets transmitted within time t, and the determination of t has great influence on the identification rate as well as the computation performance. If t is too small, the number of packets collected is also too small to form the information which has the property of statistical characteristics. If t is too large, the packets collected will take up a significant amount of system resources. We tested the value of t from 20 seconds to

300 seconds, and the identification rate of UDP sessions are presented in figure 4.

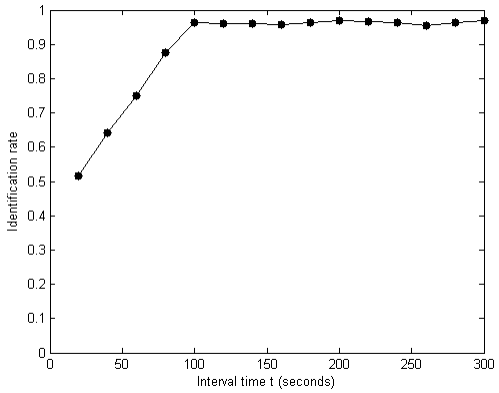


Figure 4. Identification Rate of UDP Session Using Different Time *t*

Through experiments, when *t* is larger than 100 seconds, the identification rate becomes steady; therefore, we set *t*=100 seconds as the interval time.

C. Feature Selection and Weighting Processes

According to the experiments, for TCP flow, with different number of features *d*, we obtain two results. Figure 5 shows the result without weighting method and the result with weighting method optimized by PSO-SVM algorithm.

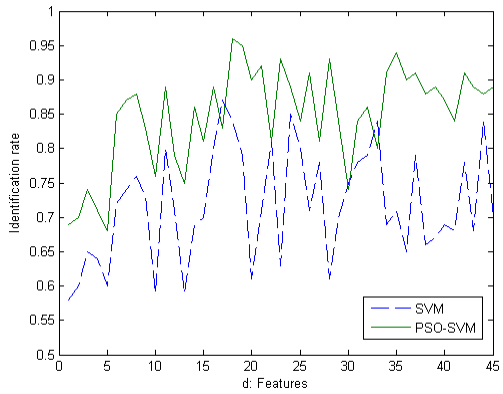


Figure 5. Identification Rate of TCP Flow by SVM

By the algorithm of GA and PSO for feature selection and weighting, for TCP traffic flow, we obtain the best recognition rate when the number of features is 18, and the largest 5 weights and their corresponding features are shown in Table 2.

For UDP traffic flow, with different number of features *d*, we also obtain two results. Figure 6 shows the result without weighting, and the result with weighting method optimized by PSO-SVM algorithm respectively.

For UDP sessions, we obtain the best recognition rate when the number of features is 25, and the largest 5 weightings and their corresponding features are shown in Table 3.

TABLE II.
THE FEATURE SUBSET AND WEIGHTS OF TCP FLOWS

Description of Features	Weights
Mean of time interval that a host receives packets in a session	1.95
Ratio of number of packets received and sent in a session	1.68
Ratio of bytes received and sent in a session	1.56
Variance of time interval that a host receives packets in a session	1.54
Number of packets a host sends in a session	1.38

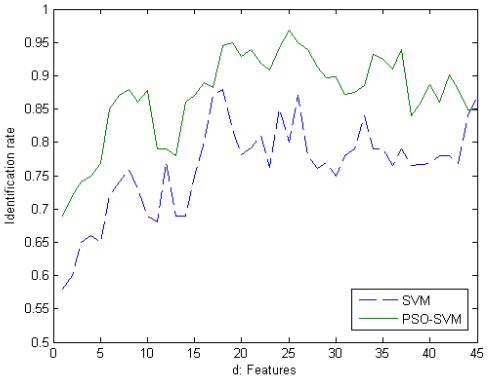


Figure 6. The identification rate of UDP flow by SVM

TABLE III.
THE FEATURE SUBSET AND WEIGHTS OF UDP FLOWS

Description of Features	Weights
Ratio of number of packets received and sent in a session	1.67
Ratio of bytes received and sent in a session	1.56
First quartile of time interval that a host receives packets in a session	1.45
Mean of the first five packets‘ time interval a host sends in a session	1.24
Variance of the first five packets‘ time interval a host sends in a session	1.18

D. Identification Rate Comparison

With the optimal value of *d*, and the corresponding weights of features selected, the identification rates of all the classes are shown in Table 4.

The average identification rate of the PSO optimized SVM algorithm, the traditional SVM algorithm without weighting method and the Bayesian neural network in [19] are listed in Table 5.

Besides choosing 50% of the data in set 1, we also choose 0.1%, 1% and 10% of the data in set 1 for testing. As we can see from Table 5 that the identification rate of PSO optimized SVM algorithm increases 12.6% compared with the traditional SVM algorithm, proving that the optimized algorithm is effective for improving

TABLE IV.
THE IDENTIFICATION RATES OF ALL CLASSES

Classification	Identification Rate
WWW	99.4±0.4%
P2P (File Sharing)	97.5±0.3%
P2P (Multimedia)	93.5±0.5%
P2P (Instant messaging)	95.5±0.7%
ATTACK	97.7±0.4%
MULTIMEDIA	95.4±0.4%
GAMES	95.2±1.1%
MAIL	99.1±0.4%
INTERACTIVE	97.8±0.3%
DATABASE	97.2±0.8%
BULK	99.1±0.5%
SERVICES	97.9±0.8%
AVERAGE	97.1±0.5%

TABLE V.
THE COMPARISON OF THREE ALGORITHMS

Training Set	Testing Set	Traditional SVM Algorithm Without Weighting	Bayesian Neural Network [19]	PSO Optimized SVM Algorithm
0.1 of Set 1	Set 2	74.4±1.5%	88.3± 0.9%	86.9±0.4%
1% of Set 1	Set 2	76.8±0.4%	91.9± 0.9%	93.2±0.8%
10% of Set 1	Set 2	82.5±0.7%	93.1± 1.3%	94.9±1.2%
50% of Set 1	Set 2	84.5±1.4%	95.3± 0.9%	97.1±0.5%

the recognition. Compared with the methods in [19], we can see that the method in this approach also improves the identification rate, and more importantly, the method proposed in this work can classify both TCP and UDP traffic flows, therefore, it is more competitive.

Ⅶ. CONCLUSION AND FUTURE WORK

Internet traffic identification is a research focus in recent years, and more and more researchers applied the algorithms in pattern recognition and machine learning to the problem of traffic identification. However, due to the lack of selection criteria of traffic attributes, some important attributes are not selected as the criterion for classification. The existing methods are mainly based on qualitative analysis of network protocols, and then choose one or more features as the criterion for classification. In this way, the essential differences among different protocols may not be found, thus influencing the recognition rate.

This novel Internet traffic identification algorithm with feature selection and weighting method based on GA and PSO-SVM proposed in this paper fills the gap in this field. We use intelligent computation methods to select the feature subset which can best classify different network applications, and then calculate the corresponding weights of each feature selected. According to the experiments, this algorithm has a high recognition rate. More importantly, this algorithm can dynamically adjust the feature subset selected and modify their corresponding weights with the change of behaviors of network applications, having the ability of adapt to the constantly changing protocols. The algorithm proposed in this paper has good scalability and availability, and it can classify both TCP and UDP traffic flows. In the future work, we will adopt some other classification algorithms such as neural network and so on. Secondly, the port and signature based approaches still have effects for some network applications, if these two methods can be combined with the statistical-based traffic classification approach, the identification performance will get better.

The original conference paper of this paper named “A Novel P2P Identification Algorithm Based on Genetic Algorithm and Particle Swarm Optimization” has been published on “Third International Symposium on Parallel Architectures, Algorithms and Programming, 2010”.

APPENDIX A

All the features of a session are described below:

Feature description
Duration of a session
Number of packets a host receives in a session
Number of packets a host sends in a session
Total number of packets in a session
Bytes of payload a host receives in a session
Bytes of payload a host sends in a session
Total number of bytes in a session
Mean of time interval that a host receives packets in a session
Mean of time interval that a host sends packets in a session
Variance of time interval that a host receives packets in a session
Variance of time interval that a host sends packets in a session
Maximum of time interval that a host receives packets in a session
Maximum of time interval that a host sends packets in a session
Minimum of time interval that a host receives packets in a session
Minimum of time interval that a host sends packets in a session
First quartile of time interval that a host receives packets in a session
First quartile of time interval that a host sends packets in a session
Median of time interval that a host receives packets in a session
Median of time interval that a host sends packets in a session
Third quartile of time interval that a host receives packets in a session
Third quartile of time interval that a host sends packets in a session
Mean of packet length that a host receives in a session
Mean of packet length that a host sends in a session
Variance of packet length that a host receives in a session
Variance of packet length that a host sends in a session
Maximum of packet length that a host receives in a session
Maximum of packet length that a host sends in a session
Minimum of packet length that a host receives in a session
Minimum of packet length that a host sends in a session
First quartile of packet length that a host receives in a session
First quartile of packet length that a host sends in a session
Median of packet length that a host receives in a session
Median of packet length that a host sends in a session
Third quartile of packet length that a host receives in a session
Third quartile of packet length that a host sends in a session
Ratio of number of packets received and sent in a session
Ratio of bytes received and sent in a session

Mean of the first five packets‘ length a host receives in a session
Mean of the first five packets‘ length a host sends in a session
Variance of the first five packets‘ length a host receives in a session
Variance of the first five packets‘ length a host sends in a session
Mean of the first five packets‘ time interval a host receives in a session
Mean of the first five packets‘ time interval a host sends in a session
Variance of the first five packets‘ time interval a host receives in a session
Variance of the first five packets‘ time interval a host sends in a session

ACKNOWLEDGMENT

This work was supported by the National 973 Key Basic Research Program under grant No.2007CB311106 and the Foundation Item of National Information Control Laboratory under grant NEUL20090101. This work is also supported by the Network and Trusted Computing Institute, College of Computer Science, Sichuan University.

REFERENCES

[1] BEN AZZOUNA Nadia, GUILLEMIN Fabrice. Impact of peer-to-peer applications on wide area network traffic: an experimental approach. Proceedings of IEEE Global Telecommunications Conference. Texas, USA, 2004: 1544-1548. doi: 10.1109/GLOCOM.2004.1378241.

[2] Jung-Tae Kim, Hae-Kyeong Park, Eui-Hyun Paik. Security issues in peer-to-peer systems. The 7th International Conference on Advanced Communications Technology. Phoenix Park, 2005: 1059-1063. doi: 10.1109/ICACT.2005.246141.

[3] S. SEN, WANG Jia. Analyzing peer-to-peer traffic across large networks. IEEE Trans on Networking, 2004, 2(2): 219-232. doi: 10.1109/TNET.2004.826277.

[4] S. SEN, O. Spatscheck, D. Wang. Accurate, scalable in-network identification of P2P traffic using application signatures. Proceedings of ACM WWW’04. New York, 2004: 512-521. doi: 10.1145/988672.988742.

[5] J. Erman, A. Mahanti, M. Arlitt, et al. Semi-supervised network traffic classification. Proceedings of the ACM SIGMETRICS, 2007. doi: 10.1145/1254882.1254934.

[6] S. Zander, T. Nguyen, G. Armitage. Self-learning IP traffic classification based on statistical flow characteristics. Passive and Active Network Measurement, 2005: 325-328. doi: 10.1007/978-3-540-31966-5_26.

[7] A. W. Moore, D. Zuev. Internet traffic classification using Bayesian analysis techniques. Proceedings of the ACM SIGMETRICS. 2005: 50-60. doi: 10.1145/1064212.1064220.

[8] A. W. Moore, K. Papagiannaki. Towards the Accurate Identification of Network Applications. Passive and Active Measurements Workshop. Boston, MA, USA. 2005: 41-54. doi: 10.1007/978-3-540-31966-5_4.

[9] P. Haffner, S. Sen, O. Spatscheck, et al. ACAS: automated construction of application signatures. Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data, ACM, New York, 2005: 192-202. doi: 10.1145/1080173.1080183.

[10] Thomas Karagiannis, Andre Broido. Transport Layer Identification of P2P Traffic. Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. Sicily, 2004: 121-134. doi: 10.1145/1028788.1028804.

[11] Ke Xu, Ming Zhang, Mingjiang Ye, et al. Identify P2P traffic by inspecting data transfer behavior. Computer Communications. 2010, 33:1141-1150. doi: 10.1016/j.comcom.2010.01.005.

[12] Holanda Filho, R. Fontenelle do Carmo, M.F. Maia. An Internet traffic classification methodology based on statistical discriminators. Network Operations and Management Symposium, 2008. NOMS: 907-910. doi: 10.1109/NOMS.2008.4575244.

[13] N. Williams, S. Zander, G. Armitages. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. SIGCOMM Computer Communication Review 36 (5)(2006): 5-16. doi: 10.1145/1163593.1163596.

[14] X. Lu, H. Duan, X. Li. Identification of P2P traffic based on the content redistribution characteristic. Communications and Information Technologies 2007: 596-601. doi: 10.1109/ISCIT.2007.4392088.

[15] Sebastian Zander, Thuy Nguyen. Automated traffic classification and application identification using machine learning. Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary. Sydney, NSW, 2005:250-257. doi: 10.1109/LCN.2005.35.

[16] Alice Este, Francesco Gringoli, Luca Salgarelli. Support Vector Machines for TCP Traffic classification. Computer Networks, 2009, 53: 2476-2490. doi: 10.1016/j.comnet.2009.05.003.

[17] Zhu Li, Ruixi Yuan, Xiaohong Guan. Accurate Classification of the Internet Traffic Based on the SVM Method. IEEE International Conference on Communications. Glasgow, 2007: 1373-1378. doi: 10.1109/ICC.2007.231.

[18] RAAHEMI B, Weicai Zhong, Jing Liu. Peer-to-peer traffic identification by mining IP layer data streams using concept-adapting very fast decision tree. Proc of the 20th IEEE International Conference on Tools with Artificial Intelligence. Dayton, OH, 2008: 525-532. doi: 10.1109/ICTAI.2008.12.

[19] Tom Auld, Andrew W. Moore, Stephen F. Gull. Bayesian Neural Networks for Internet Traffic Classification. IEEE TRANSACTION ON NEURAL NETWORKS, 2007, 18(1): 223-239. doi: 10.1109/TNN.2006.883010.

[20] Runyuan Sun, Bo Yang, Lizhi Peng, et al. Traffic classification using probabilistic neural networks. Sixth International Conference on Natural Computation (ICNC). Yantai, 2010: 1914-1919. doi: 10.1109/ICNC.2010.5584648.

[21] Arthur Callado, Judith Kelner, Djamel Sadok, et al. Better network traffic identification through the independent combination of techniques. Journal of Network and Computer Applications, 2010, 33: 433-446. doi: 10.1016/j.jnca.2010.02.002.

[22] J.Kennedy, R.Eberhart. Particle swarm optimization. Proceeding of IEEE International Conference on Neural Networks. Perth, 1995: 1942-1947. doi: 10.1109/ICNN.1995.488968.

[23] Te-Sheng Li, Chih-Ming Hsu. Parameter optimization of sub-35 nm contact-hole fabrication using particle swarm optimization approach. Expert Systems with Applications. 37 (2010): 878-885. doi: 10.1016/j.eswa.2009.05.077.

[24] Cheng-Jian Lin, Jun-Guo Wang, Chi-Yung Lee. Pattern recognition using neural-fuzzy networks based on improved particle swarm optimization. Expert Systems with Applications 36 (2009): 5402-5410. doi: 10.1016/j.eswa.2008.06.110.

[25] Shih-Wei Lin, Kuo-Ching Ying, Shih-Chieh Chen, et al. Particle Swarm Optimization for parameter determination and feature selection of support vector machines. Expert

Systems with Applications 35 (2008): 1817-1824. doi: 10.1016/j.eswa.2007.08.088.
[26] Sheng-wei Fei. Diagnostic study on arrhythmia cordis based on particle swarm optimization-based support vector machine. Expert Systems with Applications 37 (2010): 6784-6752. doi: 10.1016/j.eswa.2010.02.126.
[27] Hamdani T M, Alimi A M, Karray F. Distributed genetic algorithm with bi-coded chromosomes and a new evaluation function for features selection. Proceedings of the IEEE Congress on Evolutionary Computation. Canada. IEEE, 2006:581-588. doi: 10.1109/CEC.2006.1688362.

BIOGRAPHIES

Jun Tan was born in Chongqing (China) on June 1985. He received the B.S. and M.S. degrees from Sichuan University, China, in 2007 and 2010 respectively, and he is currently working toward the Ph.D. degree in School of Computer Science, Sichuan University, China. His research interests mainly focus on computer network, information security, data mining and machine learning.

Xingshu Chen received the M.S. and Ph.D. degrees from Sichuan University, China, in 2000 and 2004 respectively. She is currently a Professor and PhD Supervisor in the School of Computer Science, Sichuan University, China. Her interests are in the areas of computer network, information security and next generation Internet.

Min Du received the B.S. degree from Sichuan University, China, in 2009. He is currently working toward the Ph.D. degree in School of Computer Science, Sichuan University, China. His research interests mainly focus on computer network and information security, especially in P2P technology.

Efficient and Scalable Parallel Algorithm for Sorting Multisets on Multi-core Systems

Cheng Zhong, Zeng-Yan Qu, Feng Yang, Meng-Xiao Yin, Xia Li
School of Computer and Electronics and Information, Guangxi University, Nanning, China
E-mail: chzhong@gxu.edu.cn, quzengyan@163.com, yf@gxu.edu.cn, ymx@gxu.edu.cn, lixiaxia@163.com

***Abstract*—By distributing adaptively the data blocks to the processing cores to balance their computation loads and applying the strategy of "the extremum of the extremums" to select the data with the same keys, a cache-efficient and thread-level parallel algorithm** for s**orting Multisets on the multi-core computers is proposed. For the sorting Multisets problem, an aperiodic multi-round data distribution model is presented, which the first round scheduling assigns data blocks into the slave multi-core nodes according to the given distribution order and the other rounds scheduling will distribute data blocks into the slave multi-core nodes by first request first distribution strategy. The scheduling technique can ensure that each slave node can receive the next required data block before it finishes sorting the current data block in its own main memory. A hybrid thread-level and process-level parallel algorithm for sorting Multisets is presented on the heterogeneous cluster with multi-core nodes which have different amount of processing cores, different computation and communication capabilities and distinct size of main memory. The experimental results on the single multi-core computer and the heterogeneous cluster with multi-core computers show that the presented parallel sorting Multisets algorithms are efficient and they obtain good speedup and scalability.**

***Index Terms*— Multisets Sorting, Parallel algorithms, Multi-core computers, Heterogeneous clusters, Multi- level cache, Shared L2 cache, Thread-level parallelism, Selection, Aperiodic multi-round distribution**

I. INTRODUCTION

Some classical parallel sorting algorithms on various computing models were introduced in [1]. Recently, many researchers pay more and more attention to parallel sorting algorithms with GPU processors and multi- core processors. Purcell et al [2] developed a bitonic parallel sorting algorithm by the advantages of high computation-intensive and high bandwidth for GPU processors. Algorithm GPUTerasort [3] executes sort operations by SIMD instructions to obtain data parallelism, and it can lower memory latency and improve cache access efficiency by overlapped pointer and fragment processor memory accesses. Greb and Zachmann [4] applied a bitonic tree to rearrange data to reduce the number of comparing data, and presented a GPU-Abisort algorithm. Sintorn and Assarsson [5] proposed a hybrid parallel GPU-Sorting algorithm, which achieves higher speedup.

Cell machine is a kind of processors adopting heterogeneous multi-core architecture frame, it has a master processing core Power PC and eight SIMD processing cores. Gedik et al [6] proposed a so-called Cellsort three-level parallel merge sort algorithm on the Cell BE machine, the first level of which is an optimized SIMD bitonic sort algorithm, the second level is an optimized in-core bitonic merge algorithm and the third level is responsible of sorting a large amount of the data. Since algorithm Cellsort wanted to access the data in main memory several times, its obtained speedup is small. Keller and Kessler [7] improved algorithm Cellsort and presented a method to merge the data stored on the continued level nodes of the merge tree such that the merged sequences were not written to the main memory, but they were directly outputted to SIMD processing elements for next merge. Ramprasad and Baruah [8] designed an optimized radix sort algorithm which can ensure the load balance among the processing cores on the Cell Broadband Engine. Sharma et al [9] implemented six parallel algorithms for sorting integers on Cell processors and analyzed their speedups. Inoue et al [10] designed a so-called AASort parallel sorting algorithm on the multi-core SIMD processors, which can utilize fully SIMD instructions and multi-core thread-level parallelism, and eliminate the nonalignment memory access. Because algorithm AASort must access main memory several times in the merge sorting stage, it obtained small speedup. Chhugani and Macy [11] implemented efficiently a sorting algorithm on multi- core SIMD CPU architecture, which can reduce cache access latency by multi-core characteristics, increase computational density by vectored SIMD technique, balance the loads among multi-core processors in use of data decomposition, and eliminate the restriction of bandwidth by using multi-way merge. Li and Zhong et al [12] presented a multi-round scheduling technique for divisible loads on the heterogeneous cluster systems with multi-core computers. Zhong and Qu et al [13] proposed an aperiodic multi-round data distribution strategy and a parallel Multisets sorting algorithm on the heterogeneous clusters with multi-core computers.

According to the property of input Multisets sequence

Manuscript received December 26, 2010; revised February 20, 2011; accepted March 28, 2011.
Corresponding author: chzhong@gxu.edu.cn

doi:10.4304/jcp.7.1.30-41

and multi-core architectures, this paper extends our work [13] to study further the Multisets sorting problem on the multi-core systems, propose the efficient and scalable parallel sorting Multisets techniques on single multi-core computer and the heterogeneous cluster systems with multi-core computers respectively. The remainder of this paper is organized as follows. By considering impact of multi-level caches, different number of processing cores and parallel threads, section 2 first presents a cache-efficient and thread-level parallel algorithm for sorting Multisets on single multi-core computer, and then proposes a novel aperiodic multi-round data distribution model by applying the divisible loads principle [14] and design a hybrid process-level and thread-level parallel sorting Multisets algorithm on the heterogeneous clusters with multi-core nodes. Section 3 evaluates the execution time, speedup and scalability of the presented algorithms on single multi-core computer and the heterogeneous clusters with multi-core computers, respectively. Section 4 concludes the paperand gives further research direction.

II. PARALLEL SORTING MULTISETS ON MULTI-CORE SYSTEMS

Multisets is a special input data sequence and Sorting Multisets [15,16] is one kind of the data sorting issues.

Definition 1 Multisets is a special data sequence of size n, which includes only k distinct keys, $0<k<<n$.
For example, let *nl_s* be a set which is consists of people's age, *nl_s* is one kind of the dataset Multisets.

Definition 2 Sorting Multisets is to sort the data sequence for Multisets.

A. Cache-Efficient Thread-Level Parallel Sorting Multisets on Single Multi-core Computer

Definition 3 The so-called extremums are defined as the maximum and minimum in a maximal-minimal element sequence which is composed of all the maximal and minimal elements in the given several data sequences.

By applying the two-tiered mode, we design a cache-efficient and thread-level parallel sorting Multisets algorithm, short for algorithm CETLPS- Multisets, on single multi-core computer. The first tier of algorithm CETLPS-Multisets is a local parallel sorting on the shared L2 cache, it sorts the data subsequence whose size is proportional to the usage size of the shared L2 cache. The second tier is an efficient thread-level parallel algorithm to merge the sorted subsequences in order to achieve better communication performance among the processing cores. To reduce the time to exchange data between main memory and the shared L2 cache on the multi-core computer and decrease the time to merge the sorted subsequences, algorithm CETLPS-Multisets selects the data with the same keys by picking up recursively the extremums in the subsequences and assigns dynamically these data with the same keys to the processing cores to balance their computing loads, and it also applies the data blocking technique to minimize the time to access the main memory.

We assume that the Multisets sequence to be sorted has n data $X[1\sim n]$,the key of $X[i]$ is $X[i].key$, $i=1\sim n$.The distinct k keys in $X[1\sim n]$ are represented as K_1, K_2, …, and K_k respectively, where $0<k<<n$. The number of the data with key K_r is $se[K_r]$, where $0<se[K_r]\le n$, $r=1\sim k$, and $\sum_{r=1}^{k} se[K_r] = n$. Suppose that the multi-core computer has p processing cores, and P_j is the j-th processing core, $j=1\sim p$; the usage space of shared L2 cache can store C data, and a data block in the shared L2 cache contains M elements, where $M < C$. The input sequence for Multisets with length n is partitioned into n/M data blocks.

The memory bandwidth and cache design can influence significantly the performance of multi-core parallel algorithms. Therefore, we must minimize the time to access the main memory to design an cache-efficient and thread-level parallel sorting Multisets algorithm on multi-core computer. We divide the input data sequence for Multisets into several data blocks, which each block contains M elements. In the local sorting stage, we sort each block by the strategy of "the extremum of the extremums" and "selection", find the number of the data whose keys are equal to the maximal and minimal keys in each block respectively, and obtain the values of corresponding to the maximal and minimal keys. We merge in parallel the data with same keys by multiple threads to reduce the size of the data to be merged in the main memory and the number of exchanging data between the main memory and shared L2 cache in the merge stage.

The cache-efficient and thread-level parallel sorting Multisets algorithm on single multi-core computer, short for CETLPS-Multisets, is described as follows.

Algorithm CETLPS-Multisets
Begin
(1) $i=1$.
(2) The i-th data block with M elements from the main memory is read into the shared L2 cache. The data block in shared L2 cache are divided into p subsequences and they are distributed to p processing cores, which each processing core receives one subsequence. Each processing core sorts its subsequence and then writes the sorted subsequence to its local L1 cache.
(3) Processing core P_j executes the optimal selection algorithm using multiple threads to find the maximum max_j and minimum min_j from the keys of the sorted subsequence in its local L1 cache, and P_j writes max_j and min_j to its local L1 cache, $j=1\sim p$.
(4) One processing core executes the optimal selection algorithm using multiple threads to find maximum MAX from $\{max_1,max_2,\ldots, max_p\}$ and minimum MIN from $\{min_1,min_2,\ldots,min_p\}$, and then writes MAX and MIN to the shared L2 cache.
(5) Processing core P_j partitions the sorted subsequence in its local L1 cache into three parts $XL[j]$, $XM[j]$ and $XE[j]$,where the key of each element in $XL[j]$ is equal to MIN, the key of each element in $XM[j]$ is equal to MAX, and the key of each element in $XE[j]$ is not equal to MIN and not equal to MAX, $j=1\sim p$. MIN is

represented by *v_min*[*i*], the number of the data with key *MIN* in all the subsequences on *p* processing cores is denoted by *c_min*[*i*]. Similarly, *MAX* is represented by *v_max*[*i*], the number of the data with key *MAX* in all the subsequences on *p* processing cores is denoted by *c_max*[*i*]. *v_min*[*i*], *c_min*[*i*],*v_max*[*i*] and *c_max*[*i*] are written to the shared L2 cache, *j*=1~*p*.

(6) *p* processing cores combine *XE*[1], *XE*[2],…, and *XE*[*p*] in their local L1 caches into a sequence *XE*, and *XE* is stored on the shared L2 cache.

(7) *i*=*i*+1. if $i \leqslant n/M$ then goto step (2).

// The following is to merge data in shared L2 cache

(8) for *i*=1 to $\lceil k/2 \rceil n/M$ do

do steps (8.1),(8.2),(8.3) and (8.4) in parallel

(8.1) *v*[*i*]←*v_min*[*i*];

(8.2) *c*[*i*]←*c_min*[*i*];

(8.3) $v[2n\lceil k/2 \rceil / M - i + 1] \leftarrow v\_max[i]$;

(8.4) $c[2n\lceil k/2 \rceil / M - i + 1] \leftarrow c\_max[i]$;

endfor

// The following is to merge data in the main memory

(9) for *i*=1 to $2n\lceil k/2 \rceil / M$ do

for *j*=1 to *c*[*i*] do

if *i*=1 then *s*←0 else *s*←*s*+*c*[*i*-1];

X[*s*+*j*].*key*←*v*[*i*];

endfor

endfor

End.

For each iteratiive execution in the local parallel sorting stage, only one block is read into the shared L2 cache, the blocks are not overlapped each other. Therefore, any one element for Multisets is read from and written to the main memory only one time. On the other hand, we can obtaine the data with same keys in a data block and their amount in the local paralle sorting stage. In the parallel merge stage, according to the obtained data with same keys and their amount, the size of the data to be merged can be greatly decreased and the time to exchange data between the main memory and the shared L2 cache can be remarkably reduced. Hence, we can merge efficiently the data by parallel miltiple threads. In other words, CETLPS-Multisets is a cache-efficient thread-level parallel algorithm for sorting Multisets.

B. Fast and Scalable Parallel Sorting Multisets on Heterogeneous Clusters with Multi-core Nodes

We suppose that the heterogeneous multi-core cluster consists of one master node with multi-core processor and *d* slave nodes with multi-core processors, and its logical topology is star structure. Master node D_0 is responsible for receiving *n* input data for Multisets including only *k* distinct keys and distributing these data to *d* slave nodes, *k*<<*n*. Master node D_0 communicates with other slave nodes in serial transfer mode. In other words, master multi-core node is allowed to send data to only one slave multi-core node at the same moment, and only one slave node is permitted to return the sorted subsequence to the master node at the same time. We also assume that both of master node and slave nodes can execute simultaneously computation and communication. That is to say, master node can merge its received sorted subsequences while it sends data block to slave nodes or receives the sorted subsequence from slave nodes, and each slave node can sort its received data while it receives the data from master node or returns the sorted subsequence to master node.

For convenience, let L_i denote a communication link from master node D_0 to slave node D_i, $nLat_i$ represent the required time to start one communication in link L_i. We suppose that the wanted time to transfer a datum is $1/B_i$ and the required time to transfer N_i data is $nLat_i+N_i/B_i$ in link L_i, where B_i denotes the transfer rate and $cLat_i$ denotes the required time that slave node D_i starts a computation, the required average time that slave node D_i sorts a datum is $1/S$ and the required time that slave node D_i sorts N_i data is $cLat_i+N_i/S_i$, where S_i represents the computation rate, *i*=1~*d*. *BUF* represents the total capacity of the main memories on *d* slave nodes, size *n* of the input data for Multisets is not greater than *BUF*, and size N_i of the data distributed to slave node D_i is not greater than capacity buf_i of the main memory on slave node D_i, *i*=1~*d*.

Let C_{max} be the execution time to sort completely the input sequence for Multisets of size *n* on the heterogeneous clusters with multi-core nodes.

B.1. Single-round Data Distribution Model and Parallel Sorting for Multisets

Figure 1 describes the procedure that master node distributes data blocks for Multisets to the slave nodes using single-round distribution mode and the slave nodes sort their received data blocks in parallel on the heterogeneous clusters with multi-core nodes.

From Figure 1 we can see that for the large-scale input sequence for Multisets, if master node distributes data blocks to slave nodes in use of single-round data distribution mode, there will be two limitations. The first one is that the size of the data distributed to the salve node may be more than its storage capability. The second one is that the amount of data scheduled to the salve nodes must be increasing each time, and the computation and communication execution can not beimplemented in parallel. In other words, the required communication time will be increased. Therefore, we must study to present a more efficient data distribution strategy to sort in parallel the input Multisets sequence.

B.2. Aperiodic Multi-round Data Distribution Model and Parallel Sorting for Multisets

To design an efficient and scalable parallel algorithm for sorting Multisets on the heterogeneous clusters with multi-core machines, the issues to be solved are to how to partition properly the input data into several data blocks, how to ensure that each salve node can carry out sufficiently its computational capability and master node can communicate efficiently with slave nodes, and how to insure that each multi-core node can overlap computation and communication operation.

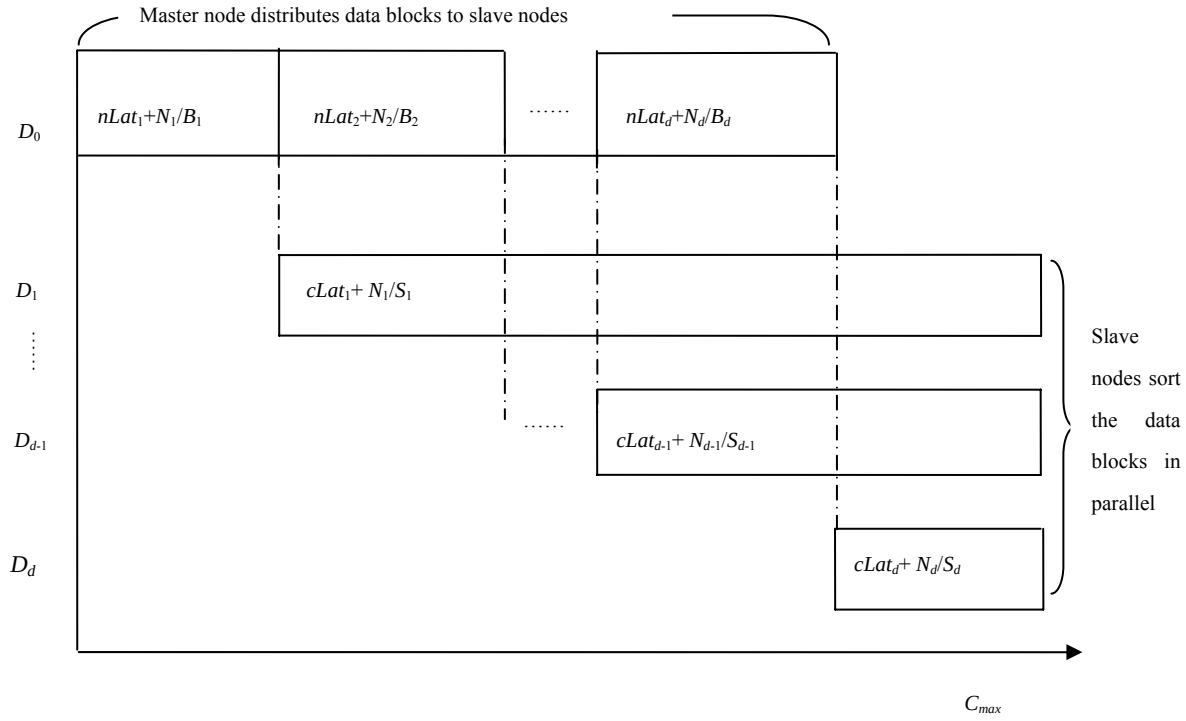


Figure 1 Procedure of distributing data with single-round and parallel sorting Multisets on heterogeneous multi-core clusters

Secondly, each salve node on the heterogeneous multi-core cluster has different size of main memory space. For the large-scale input data for Multisets, if master node distributes all the input data into the slave nodes in single round scheduling mode, the amount of the data to be assigned to each slave node may exceed its usable memory capacity, and the size of the data scheduled to each slave node will be remarkably increased. At this time, it is difficult to overlap computation and communication operations, and the communication overhead will be enhanced.

Finally, the input sequence for Multisets may be not uniform distribution. That is to say, some salve nodes will receive the data sequence for Multisets with fewer distinct keys, and the others will receive the data sequence for Multisets with relatively more different keys. Clearly, for the input sequence for Multisets with the same size, the required time to sort the data sequence with fewer distinct keys is much less than the required time to sort the data sequence with more different keys. If master node distributes the input data for Multisets to the slave nodes in each round scheduling mode according to their computational capability only, different slave node may require distinct amount of time to sort completely its received data sequence. At this time, some slave nodes are busy and the others will be free.

To solve the above problems, we present an aperiodic multi-round data distribution strategy to sort the input sequence for Multisets. Its goal is to ensure that all the slave nodes are always busy and each slave node can sort completely its received data sequence at the same time. The presented aperiodic multi-round data distribution strategy must guarantee that each slave node can receive the next data block before it has sorting the current data block, each slave node immediately starts to sort the next data block when it has sorted the current data block, and there is no free waiting time in the parallel sorting process. The free waiting will only appear when no slave nodes request master node to send data block and master node merges completely the sorted subsequences returned from the slave nodes.

Definition 4 The so-called First Request First Distribution (short for FRFD) strategy is that master node is responsible for storing the input data sequence and distributing the data to the slave multi-core nodes on the heterogeneous cluster in multi-round scheduling mode, and if over a salve node require the next data blocks, then master node will distribute the data blocks to the slave node with the first request.

Now we present an aperiodic multi-round data distribution model on the heterogeneous clusters with multi-core nodes:

$$\sum_{i=1}^{d} N_i = n \tag{1}$$

$$N_i \geq 0, 1 \leq i \leq d \tag{2}$$

$$cLat_i + N_i / S_i + \sum_{j=1}^{i} nLat_j + N_j / B_j \leq C_{\max}, \quad 1 \leq i \leq d \tag{3}$$

$$N_i \leq buf_i, 1 \leq i \leq d \tag{4}$$

$$\sum_{i=1}^{d}(nLat_i + bk_i / B_i) \le tim \quad (5)$$

$$bk_i < buf_{\min} / 2, i = 1 \sim d \quad (6)$$

$$\sum_{i=1}^{d} bk_i < n \quad (7)$$

where *tim* represents the required execution time that the slave node with the strongest computational ability sorts the subsequence of size bk_i, i=1~d, and $buf_{\min}$ denotes the capacity of main memory on the slave node with the smallest storage capability. The value of bk_i will be determined by the experiment, i=1~d. We set up bk_i=buf_i /j^l and can obtain the appropriate value of bk_i to meet the conditions (5), conditions (6) and conditions (7) by experimental testing, where j is a positive integer and j>2, l is also a positive integer and l≥1, i=1~d.

The parallel sorting Multisets algorithm using the aperiodic multi-round data distribution strategy on the heterogeneous clusters with multi-core nodes, short for PSAMRD-Multisets, is described as follows:

Algorithm 2 PSAMRD-Multisets

Begin

(1) According to the given order of nodes D_1, D_2,..., and D_d, master node D_0 distributes two data blocks to slave node D_i, which the size of each block is bk_i, i=1~d.

(2) Slave node D_i immediately sorts the first received data block of size bk_i by algorithm CETLPS-Multisets while it has received two data blocks, i=1~d.

(3) After the first received data block has been sorted, slave node D_i immediately begins to sort the second received data block and retruns the sorted data subsequence of size bk'_{ji} to master node D_0, and slave node D_i simuitaneously requests master node D_0 to distribute the next data block of size bk_i to it, i=1~d. Since slave node D_i returns only the data with same key in the sorted subsequence of size bk_i and the amount of these data to master node D_0, we can know that bk'_{ji}<<bk_i, i=1~d.

(4) If more than one slave node requests master node D_0 to distribute the next data block, master node D_0 sends the next data block of size bk_i to the corresponding slave node by the First Request First Distribution strategy. Repeat this step until all the data blocks for Multisets have been distributed and sorted.

(5) Master node D_0 merges the sorted subsequences returned from d slave nodes to a new sorted longer sequence while it distributes the data blocks to the slave nodes in each round scheduling. Hence, node D_0 can almost merge completely the sorted subsequences to a whole sorted sequence of length n when d slave nodes have sorting their received data blocks.

End.

Figure 2 shows that master node D0 distributes the data blocks to the slave nodes and the slave nodes sort the received data blocks and return the sorted subsequences to master node on heterogeneous cluster with one master multi-core node and three slave multi-core nodes by using the aperiodic multi-round scheduling and first request first distribution strategy.

For the presented algorithm PSAMRD-Multisets, we emphasizes sufficiently utilization of both heterogeneity of the multi-core clusters and the multi- level cache and multiple threading techniques. Algorithm PSAMRD-Multisets has the following advanteges. Firstly, it considers the usable amount of main memory capacity on each slave node and distributes the data block with appropriate size to each slave node. Secondly, it aims at the property of sorting Multisets, it ensures that each slave node can receive the next data block befor it has sorting the current data block and the communication links on the hetergeneous multi-core cluster are alway busy, and it can improve remarkably the performance of parallel sorting Multisets. Thirdly, it applys first request first distrbution strategy to assign the data blocks to the slave nodes, and it can eliminate the restriction of the bandwidth of main commuincation link. Fourthly, it applies the aperiodic multi-round scheduling strategy to distribute data blocks to the slave nodes, and it sorts quickly the data blocks on the single multi-core node by executing algorithm CETLPS-Multisets and can overlap computation and communication operation. Finally, we notice that the input sequence for Multisets of length n has only distinct k keys, where k<<n, and since each slave node returns only the sorted data with same key and the amount of these data to master node while it has sorting its received data subsequence, the amount of the data returned to master node is very small and the required communication overhead can be ignored.

III. EXPERIMENT

We first test the cache-efficient and thread-level parallel sorting Multisets algorithm on single multi-core computer, then evaluate the parallel sorting Multisets algorithm using aperiodic multi-round distribution strategy on the heterogeneous clusters with multi-core computers and analyze its scalability.

A. The experiments on single multi-core computer

The multi-core computer has Intel Core(TM) 2 Quad Processor, the running operating system is Linux kernel 12.0, the programming language is C and parallel communication library is OpenMP. Each element of input sequence for Multisets is from {0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,…,748.5,749,749.5,750}, and k=1501. We select a half of size of the shared L2 cache on the Intel Core(TM) 2 Quad Processor as the size of a data block.

We test the execution time and speedup of algorithm CETLPS-Multisets with different number of the processing cores and distinct number of parallel threads, and evaluate the performance of algorithm CETLPS-Multisets with blocking and non-blocking data technique, respectively.

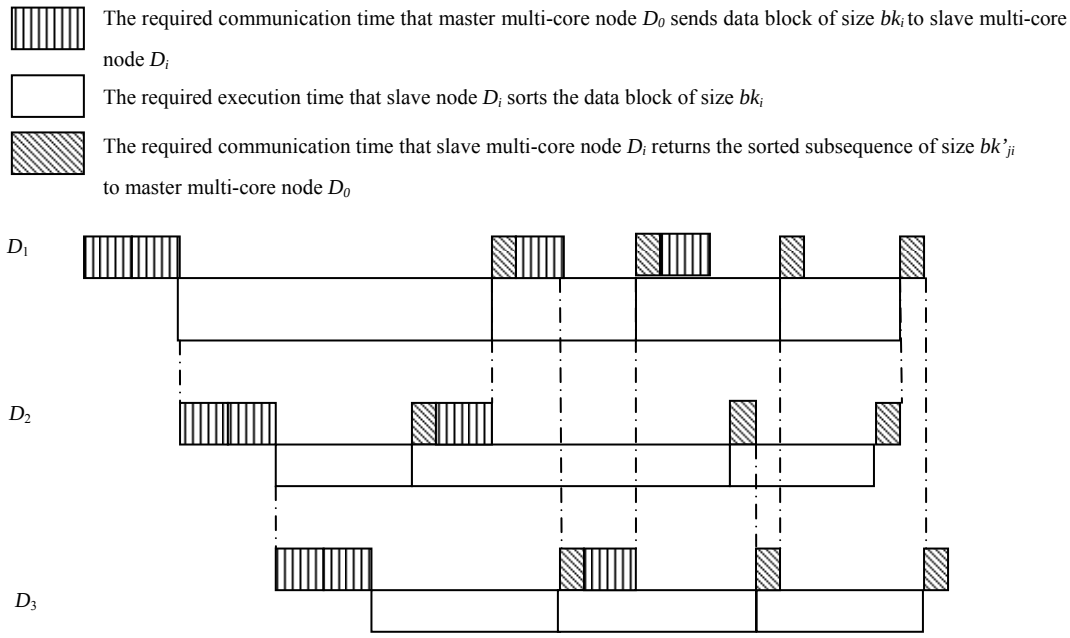


Figure 2. Procedure that master node distributes the data to slave nodes, slave nodes sort the received data and return the sorted subsequences to master node on heterogeneous multi-core clusters

By switching dynamically the processing cores on the multi-core computer, we test the performance of algorithm CETLPS-Multisets.

Figure 3 shows the execution time of algorithm CETLPS-Multisets using four processing cores and different number of parallel threads respectively.

From Figure 3, we can see that for the input data of different size, when algorithm CETLPS-Multisets is executed by 4 processing cores, its required time is gradually decreased with the number of threads varying from 4 to 8, and the required time is gradually increased with the number of threads varying from 8 to 20. In other words, if algorithm CETLPS-Multisets is executed by 4 processing cores and 8 threads, its required time is the smallest.

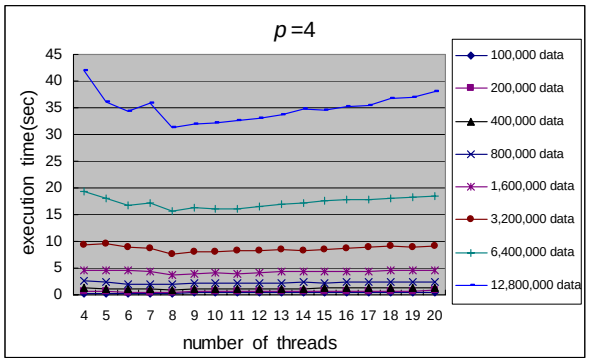


Figure 3 Execution time of algorithm CETLPS-Multisets using 4 processing cores and multiple threads

Figure 4 gives the execution time of algorithm CETLPS-Multisets using three processing cores and different number of threads respectively.

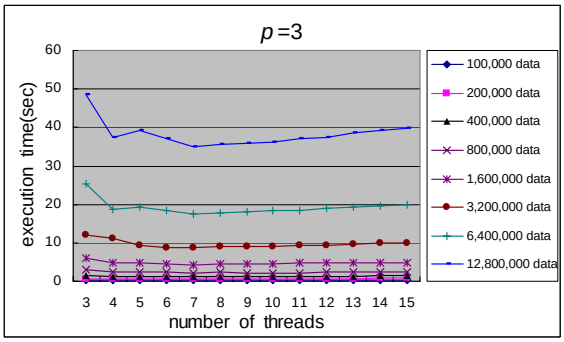


Figure 4 Execution time of algorithm CETLPS-Multisets using three processing cores and multiple threads

Figure 4 shows that for the input data with different size, when algorithm CETLPS-Multisets is executed by 3 processing cores, its execution time is gradually decreased with the number of threads varying from 3 to 7, and the execution time is gradually increased with the number of threads varying from 7 to 15. Hence, if algorithm CETLPS-Multisets is executed by 3 processing cores and 7 threads, its execution time is the smallest.

Figure 5 displays the execution time of algorithm CETLPS-Multisets using two processing cores and multiple threads.

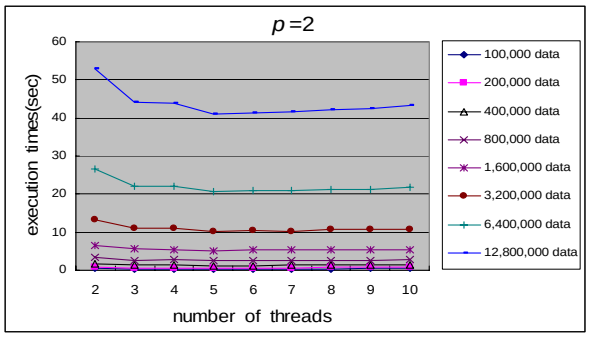


Figure 5 Execution time of algorithm CETLPS-Multisets using 2 processing cores and multiple threads

According to Figure 5, for the input data with different size, when algorithm CETLPS-Multisets is executed by 2 processing cores, its execution time is gradually decreased with the number of threads varying from 2 to 5, and the execution time is gradually increased with the number of threads varying from 5 to 10. It represents that its execution time is the smallest if algorithm CETLPS-Multisets is executed by two processing cores and five threads.

When the size of input data is fixed, Figure 6 shows the speedup of algorithm CETLPS-Multisets using four processing cores and multiple threads.

As we can see from Figure 6 that, when the size of input data is 12,800,000, if algorithm CETLPS- Multisets is executed by 4 processing cores and 8 threads, its speedup is the largest, and if algorithm CETLPS-Multisets is executed by 4 processing cores and 4 threads, its speedup is the smallest.

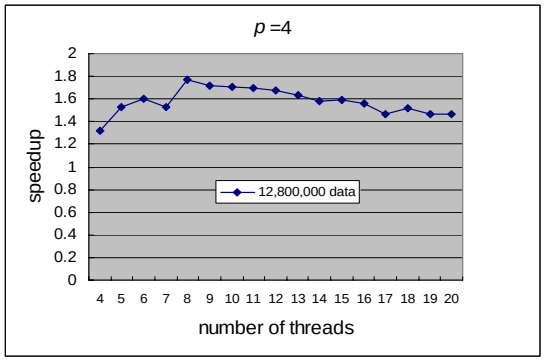


Figure 6 Speedup of algorithm CETLPS-Multisets using 4 processing cores and multiple threads

For the size of input data is 12,800,000, Figure 7 illustrates its speedup when algorithm CETLPS-Multisets is executed by three processing cores and multiple threads.

As we have shown in Figure 7, when the size of input data is 12,800,000, if algorithm CETLPS- Multisets is executed by three processing cores and seven threads, it obtains the largest speedup, and if algorithm CETLPS-Multisets is executed by three processing cores and three threads, it achieves the smallest speedup.

For the size of the input data is 12,800,000, when algorithm CETLPS-Multisets is executed by two processing cores and multiple threads, Figure 8 gives its obtained speedup.

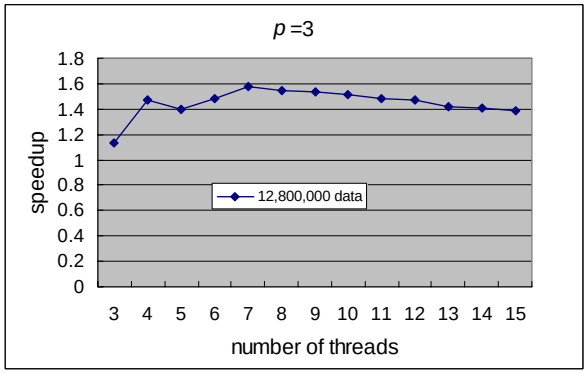


Figure 7 Speedup of algorithm CETLPS-Multisets using 3 processing cores and multiple threads

From Figure 8, we can see that when the size of input data is 12,800,000, its speedup is the smallest if algorithm CETLPS-Multisets is executed by two processing cores and two threads, and its speedup is the largest if algorithm CETLPS-Multisets is executed by two processing cores and five threads.

Figure 9 shows the execution time of algorithm CETLPS-Multisets using the threads of the optimal number and multiple processing cores.

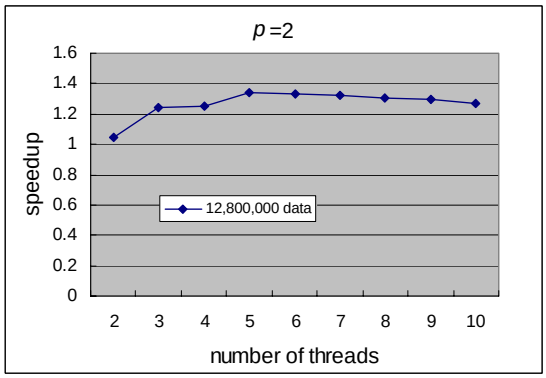


Figure 8 Speedup of algorithm CETLPS-Multisets using two processing cores and multiple threads

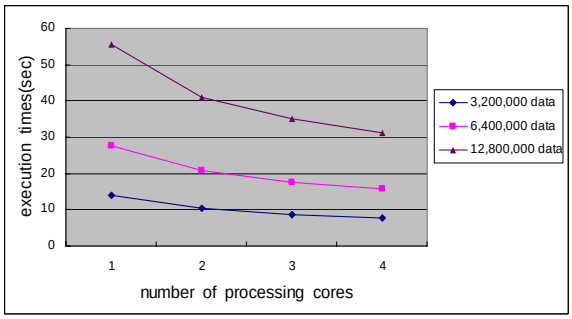


Figure 9 Execution time of algorithm CETLPS-Multisets using multiple processing cores and threads of optimal number

From Figure 9 we can see that, for the input data of different size, if algorithm CETLPS-Multisets is executed by threads with the optimal number, its required time is gradually decreased with the increase of used processing cores. Furthermore, if algorithm CETLPS-Multisets is executed by more processing cores, the descending trend of its execution time is gradually slow. This is due to the fact that the proportion of the communications among the

processing cores is large at this time, at the meanwhile, several processing cores may want to access some shared resources and spend some additional time.

Figure 10 gives the speedup of algorithm CETLPS-Multisets when it is executed by multiple processing cores and the threads of optimal number.

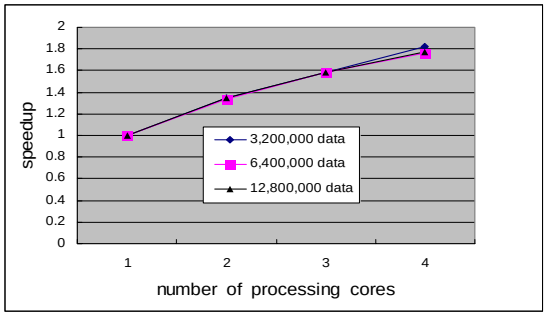


Figure 10 Speedup of algorithm CETLPS-Multisets using multiple processing cores and threads with optimal number

From Figure 10, we can see that the speedup of algorithm CETLPS-Multisets is gradually ascended with increase of used processing cores. However, when algorithm CETLPS-Multisets is executed by using more processing cores, there are more and more communications on the processing cores and the ascending trend of speedup is relatively slow.

When algorithm CETLPS-Multisets with data blocking and non-blocking is executed by 1, 2, 3 and 4 processing cores, respectively, Figure 11 shows their execution time. When algorithm CETLPS-Multisets is implemented by blocking data technique, the size of each data block in main memory is designated by the algorithm and then the data block is read into the shared L2 cache. When algorithm CETLPS-Multisets is implemented by non-blocking data technique, the size of each data block in main memory is designated by the computer system and then the data block is read into the shared L2 cache.

As shown in Figure 11, the required time of executing algorithm CETLPS-Multisets with blocking data is obviously less than that of the one with non-blocking data, and the blocking data technique is more suitable for parallel sorting the large-scale input sequence for Multisets.

For the input dataset with various distribution, when the size of input data is fixed n=12,800,000, Figure 12 gives the execution time of algorithm CETLPS- Multisets using the threads with optimal number and multiple processing cores.

From Figure 12, we can see that when algorithm CETLPS-Multisets is executed by the threads of optimal number, its required time is gradually decreased with the increase of used processing cores. Furthermore, the required time to sort the almost sorted input dataset is the smallest, and the execution time to sort the random input dataset is the most.

For the input sequence with various distribution, Figure 13 illustrates the execution time of algorithm CETLPS-Multisets using four processing cores and the threads of optimal number respectively.

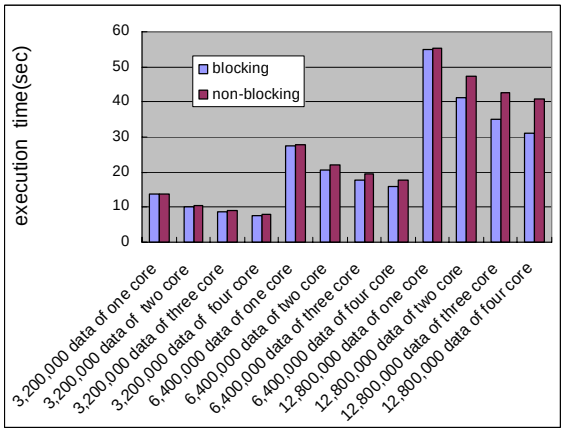


Figure 11 Execution time of algorithm CETLPS-Multisets with data blocking and non-blocking

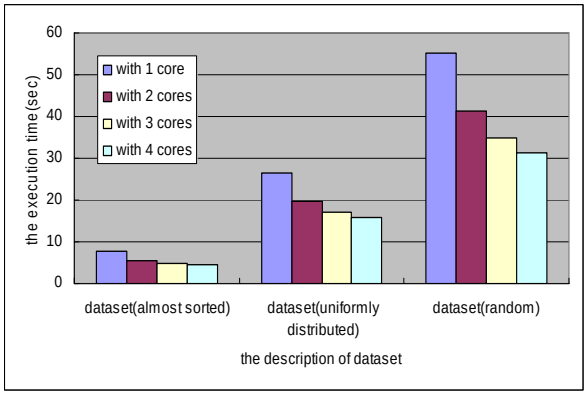


Figure 12 Execution time of CETLPS-Multisets with the input data of various distribution using multiple processing cores

We can see from Figure 13 that for the input data set with different size, when algorithm CETLPS- Multisets is executed, its required time to sort the almost sorted dataset is the smallest, and its required time to sort the random dataset is the most.

Figure 14 shows the curve of equivalent efficiency function of algorithm CETLPS-Multisets with input data of different size.

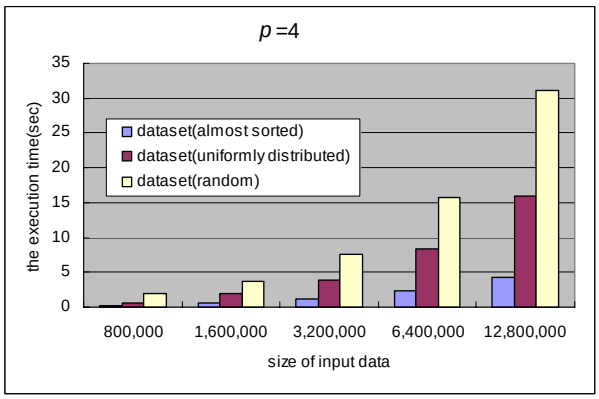


Figure 13 Execution time of algorithm CETLPS-Multisets for the input data with various distribution using 4 processing cores

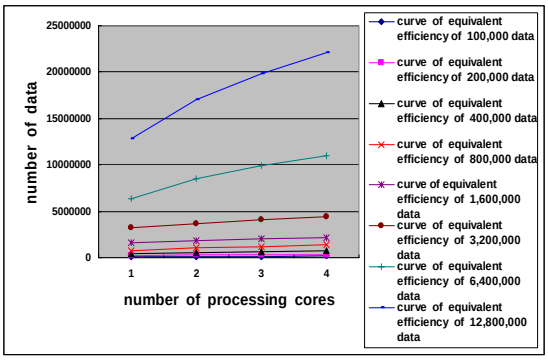


Figure 14 Equivalent efficiency function curve of algorithm CETLPS-Multisets with input data of different size

From Figure 14, we can see that the increase of the required workload is sub-linearly proportional to the increase of used processing cores to maintain the equivalent efficiency of algorithm CETLPS-Multisets for the input data of arbitrary size. Hence, algorithm CETLPS-Multisets has good scalability.

B. The experiments on the heterogeneous cluster with multi-core computers

The experimental platform is a heterogeneous cluster, which consists of the multi-core computers connected via a 100Mbps Ethernet network. The operating system is Red Hat Linux 12.0. The programming language and development environment are C, MPI and OpenMP respectively. The configuration of the heterogeneous multi-core cluster is listed in Table I. The input sequence for Multisets used in the experiment are also from the set of {0, 0.5, 1, 1.5, 2, 2.5, …, 99.5, 100, 100.5,…, 700.5,…,749.5, 750}, where k=1501.

A multi-core computer with the main memory of size 512MB and L2 cache of size 2MB is used as the master node and the other multi-core computers are treated as the slave nodes in the experiment. We test each communication link between master node and every slave node twice and obtain the relationship between the amount of data to be transferred and the required communication time, we achieve the values of B_i and $nLat_i$ by solving the corresponding equations, i=1~d. By executing twice algorithm CETLPS-Multisets on each multi-core node, we get the relationship between the amount of data to be sorted and the required computation time, and obtain the values of S_i and $cLat_i$ by solving the corresponding equations, i=1~d.

TABLE I.
CONFIGURATION OF COMPUTING NODES ON HETEROGENEOUS MULTI-CORE CLUSTER

Type of Nodes	CPU	Size of Main Memory	Size of Shared L2 Cache
INTEL	Intel quad-core	2GB	12MB
IBM	AMD dual-core	1GB	1MB
HP	Intel single-core	512MB	2MB
LEGEND	Intel dual-core	2GB	4MB

We evaluate the execution performance of algorithm PSAMRD-Multisets. Figure 15 first shows its execution time with the input data of different size and multiple multi-core computers.

From Figure 15 we can see that with the increase of the amount of multi-core nodes, the required execution time of algorithm PSAMRD-Multisets is remarkably decreased. We also see that the decrease of the execution time of algorithm PSAMRD-Multisets is slow down when more and more multi-core nodes is used. The reason is that the required communication overhead and the overhead to access shared cache and other resources will be increased when more and more multi-core nodes are used.

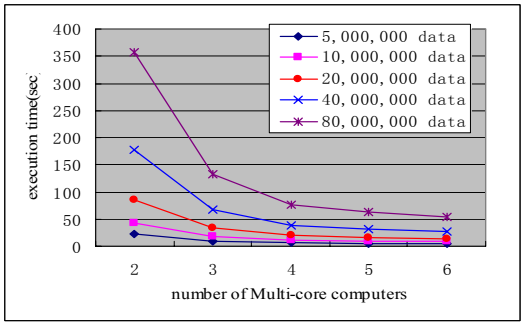


Figure 15. Execution time of algorithm PSAMRD-Multisets with input data of different size and multiple multi-core nodes

For the input sequence for Multisets of different size, the obtained speedup of algorithm PSAMRD-Multisets is given in Figure 16.

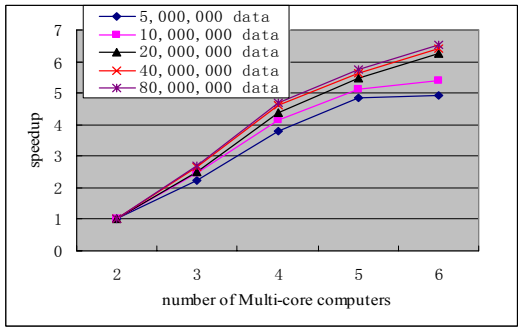


Figure 16. Speedup of algorithm PSAMRD-Multisets with input data of different size

Figure 16 shows that for the fixed number of multi-core nodes, the speedup of algorithm PSAMRD-Multisets is enhanced when the size of the input data sequence is increased. Besides, it shows that the increase of the speedup is slow down when more and more multi-core nodes are used.

When the amount of used multi-core nodes is d=6, Figure 17 displays speedup of algorithm PSAMRD-Multisets when the size of input data sequence is increased progressively.

We can see from Figure 17 that with increase of the size of the input data, the speedup of algorithm PSAMRD-Multisets is progressively enhanced, and the increase trend of the speedup is slowly to be stable. This result shows that algorithm PSAMRD-Multisets is very efficient for sorting the large-scale input data sequence for Multisets.

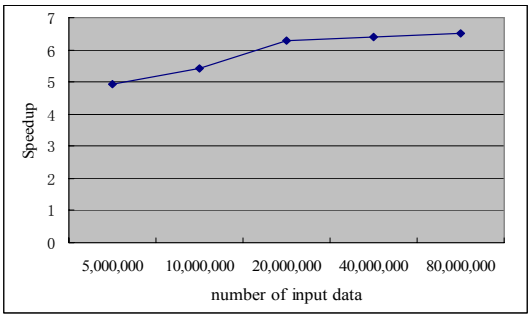


Figure 17. Speedup of algorithm PSAMRD-Multisets with input data of different size and 6 multi-core nodes

For the different combination with the amount of multi-core nodes and the number of processing cores, when the number of input data sequence for Multisets is increased, the required execution time of algorithm PSAMRD-Multisets is shown in Figure 18.

Figure 18 indicates that when the size of the input data sequence is the same, the required execution time of algorithm PSAMRD-Multisets is decreased with increase of the amount of multi-core nodes; when the size of the input data sequence is the same and the amount of used multi-core nodes is also the same, the more the nodes used, the shorter the required execution time of algorithm PSAMRD-Multisets. Figure 18 also shows that algorithm PSAMRD-Multisets can utilize sufficiently the hardware characteristics and software technology of multi-core architectures.

The scalability of the presented parallel sorting algorithm PSAMRD-Multisets can be analyzed by the equivalent efficiency functions.The curve of the equivalent efficiency function of algorithm PSAMRD-Multisets is given in Figure 19.

Figure 19 shows that for arbitrary size *n* of input data sequence, the increase of size of the required workload is sub-linearly proportional to the increase of used multi-core computers (nodes) in order to maintain the equivalent efficiency of algorithm PSAMRD-Multisets. Therefore, algorithm PSAMRD-Multisets has good scalability and it can utilize the computational ability of the increasing multi-core nodes.

IV. CONCLUSIONS

In this paper, we propose the cache-efficient, thread-level parallel and scalable algorithms for sorting Multisets on single multi-core computer and the heterogeneous cluster with multi-core computers, respectively. The presented algorithms sufficiently consider the distribution property of input dataset Multisets and the heterogeneity of multi-core clusters, utilize the function of multi-level cache and parallel multi-threading technique. The presented algorithms implement sorting Multisets using thread-level parallel selection and merge techniques. To efficiently sorting Multisets on heterogeneous multi-core clusters, this paper presents a single-round data distribution model and an aperiodic multi-round data distribution model. The presented aperiodic multi-round data distribution strategy can guarantee no free waiting time in the parallel sorting. On the one hand, the next works is study to design the cache-efficient, fast and scalable thread-level parallel sorting algorithm for more general sequence on the heterogeneous multi-core clusters. On the other hand, we will also study to design the efficient and scalable parallel sorting algorithms on the heterogeneous cluster systems with multi-core general computers and GPU computers.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China under grant NO. 60963001, Project of Outstanding Innovation Teams Construction Plans at Guangxi University, and Program to Sponsor Teams for Innovation in the Construction of Talent Highlands in Guangxi Institutions of Higher Learning.

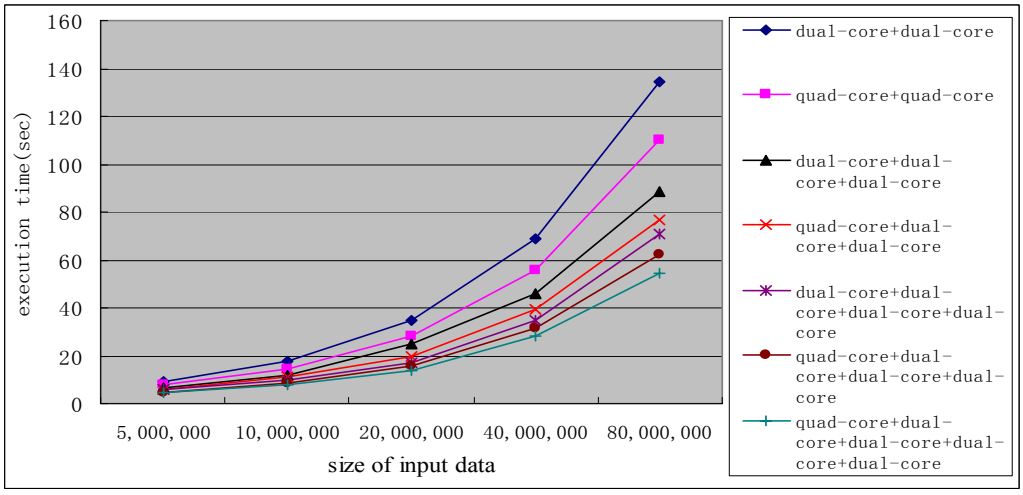


Figure 18. Execution time of algorithm PSAMRD-Multisets with various combination of amount of multi-core nodes and processing cores when the size of input data sequence is increased

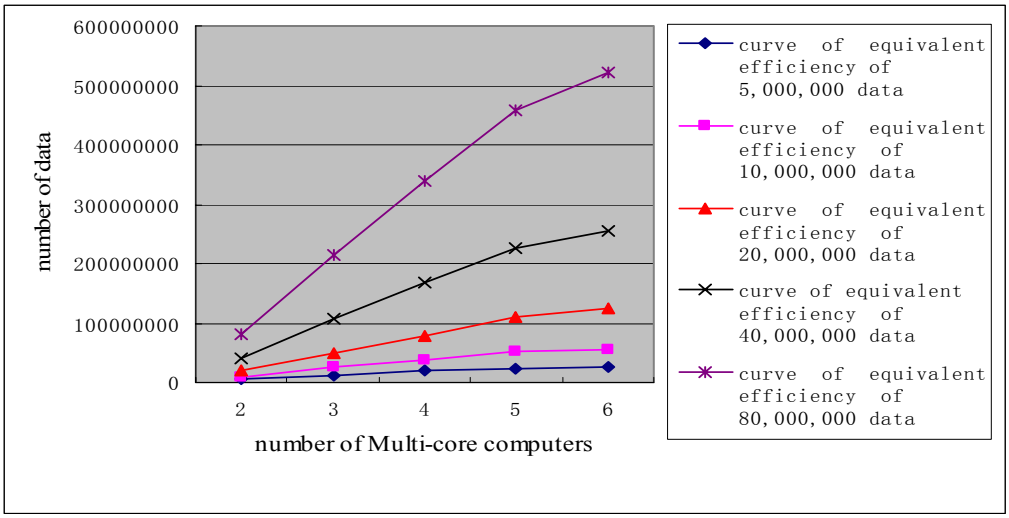


Figure 19. Curve of the equivalent efficiency function of algorithm PSAMRD-Multisets

REFERENCES

[1] Chen Guoliang, Design and Analysis of Parallel Algorithms, Third edition, Higher Education Press, Beijing,2009.

[2] T. J. Purcell, C.Donner, M.Cammarano, et al, "Photon mapping on programmable graphics hardware", Proc. of 2003 ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware, ACM Press, 2003,pp.41-50.

[3] N. Govindaraju,J. Gray,R. Kumar,et al, "Gputerasort: High performance graphics coprocessor sorting for large database management", Proc. of 2006 ACM SIGMOD /PODS, Chicago, USA, ACM Press, 2006, pp.325-336.

[4] A. Greb,G. Zachmann, "GPU-ABiSort: efficient parallel sorting on stream architectures", Proc. of 2006 Parallel and Distributed Processing Symposium(IPDPS'06), Rhodes Island, Greece, 2006, pp. 25-29.

[5] E. Sintorn,U. Assarsson, "Fast Parallel GPU-Sorting Using a Hybrid Algorithm", Journal of Parallel and Distributed Computing, Vol.68,No.10, 2008, pp.1381 -1388.

[6] B. Gedik, R. R. Bordawekar, and P. S. Yu, "CellSort: High Performance Sorting on the Cell Processor", Proc. of VLDB '07, 2007, pp. 1286-1297.

[7] J. Keller, C. W. Kessler, "Optimized Pipelined Parallel Merge Sort on the Cell BE", Proc. of 14th International European Conference on Parallel and Distributed Computing (Euro-Par), Highly Parallel Processing on a Chip (HPPC),2008,pp.26-29.

[8] N. Ramprasad, Pallav Kumar Baruah, " Radix Sort on the Cell Broadband Engine", Proc. of . 2007 Intl. Conf. on High Performance Computing (HiPC), Posters, 2007.

[9] D. Sharma,V. Thapar,R. Ammar, "Efficient sorting algorithms for the cell broadband engine", Proc. of 2008 IEEE Symposium on Computers and Communications, 2008, pp.736-741.

[10] H. Inoue,T. Moriyama,H. Komatsu, et al, "AA-Sort: A New Parallel Sorting Algorithm for Multi-Core SIMD Processors", Proc. of PACT'07, 2007, pp.189-198.

[11] J. Chhugani, W. Macy, and V. W. Lee, et al, "Efficient Implementation of Sorting on MultiCore SIMD CPU Architecture", Proceedings of the VLDB Endowment, Vol.1, No.2, 2008, pp.1313-1324.

[12] Li Xia, Zhong Cheng, Qu Zeng-Yan, "Multi-round Scheduling for Divisible Loads on the Heterogeneous Cluster Systems of Multi-core Computers", Proc. of the 2nd International Conference on Interaction Science: Information Technology, Culture and Human, ACM Press, Vol.2, 2009,pp.846-852.

[13]Cheng Zhong, Zeng-yan Qu, Feng Yang, Meng-xiao Yin, "Parallel Multisets Sorting Using Aperiodic Multi-round Distribution Strategy on Heterogeneous Multi-core Clusters", Proc. of 3rd International on Parallel Architectures, Algorithms and Programming, IEEE Computer Society Press, Dec.2010, pp.247-254.

[14] Y. Yang, Henri Casanova, "RUMR: Robust Scheduling for Divisible Workloads", Proc. of the 12th IEEE International Symposium on High Performance Distributed Computing, IEEE Computer Society Press, 2003,pp. 114-123.

[15] S. Rajasekaran, "An efficient parallel algorithm for sorting multisets", Information Processing Letters, Vol.67, No.3,1998, pp.141-143.

[16] Zhong Cheng, Chen Guo-Liang, "An Efficient Parallel Sorting Algorithm for Multisets", Journal of Computer Research and Development, Vol.40,No.2,2003,pp.335-361.

Cheng Zhong received a Ph.D. degree in department of computer science and technology at University of Science and Technology of China in July 2003. He is working as a professor of computer science in the school of computer and electronics and information at Guangxi University in China. He is a senior member of China Computer Federation (CCF), a standing member of CCF on Theoretical Computer Science Committee and Open systems Committee and PC Co-Chairs and PC member for several international and national conferences in computer science and technology. He has published over 60 papers in the journals and conferences and 7 textbooks. His current main research interests include parallel and distributed computing, network security, highly -trusted software, and Bioinformatics.

Zeng-yan Qu received a M.Sc degree in the school of computer and electronics and information at Guangxi University in China in July 2010. Her research interest is parallel and distributed computing.

Feng Yang received a M.Sc degree in school of computer and electronics and information at Guangxi University in China in July 2005. He is working as a lecturer in the school of

computer and electronics and information at Guangxi University. His research interests include parallel computing, network security, and Bioinformatics.

Meng-Xiao Yin received a M.Sc degree in school of Mathematics and information at Guangxi University in China in July 2005. She is working as a lecturer in the school of computer and electronics and information at Guangxi University. Her research interests include parallel computing and graph theory.

Xia Li received a M.Sc degree in the school of computer and electronics and information at Guangxi University in China in July 2010. Her research interest is parallel and distributed computing.

A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment

Jianhua Gu
School of Computer
NPU HPC Center
Xi'an, China
gujh@nwpu.edu.cn

Jinhua Hu
School of Computer
NPU HPC Center
Xi'an, China
jinhua@mail.nwpu.edu.cn

Tianhai Zhao
School of Computer
NPU HPC Center
Xi'an, China
zhaoth@nwpu.edu.cn

Guofei Sun
School of Computer
NPU HPC Center
Xi'an, China
sunguofei@mail.nwpu.edu.cn

***Abstract*—In view of the load balancing problem in VM resources scheduling, this paper presents a scheduling strategy on load balancing of VM resources based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy computes ahead the influence it will have on the system after the deployment of the needed VM resources and then chooses the least-affective solution, through which it achieves the best load balancing and reduces or avoids dynamic migration. At the same time, this paper brings in variation rate to describe the load variation of system virtual machines, and it also introduces average load distance to measure the overall load balancing effect of the algorithm. The experiment shows that this strategy has fairly good global astringency and efficiency, and the algorithm of this paper is, to a great extent, able to solve the problems of load imbalance and high migration cost after system VM being scheduled. What is more, the average load distance does not grow with the increase of VM load variation rate, and the system scheduling algorithm has quite good resource utility.**

***Index Terms*—computing; virtual machine resources; load balancing; genetic algorithm; scheduling strategy**

I. INTRODUCTION

Cloud computing is a new technology in academic world [1]. On cloud computing platform, resources are provided as service and by needs, and it guarantees to the subscribers that it sticks to the Service Level Agreement (SLA). However, due to the situation that the resources are shared, and the needs of the subscribers have big dynamic heterogeneity and platform irrelevance, it will definitely lead to resource waste if the resources cannot be distributed properly[2]. Besides, the cloud computing platform also needs to dynamically balance the load among the servers in order to avoid hotspot and improve resource utility. Therefore, how to dynamically and efficiently manage resources and to meet the needs of subscribers become the problems to be solved.

Virtualization technology provides an effective solution to the management of dynamic resources on cloud computing platform. Through sealing the service in virtual machines and mapping it to every physical server, the problem of the heterogeneity and platform irrelevance of subscribers' needs can be better solved and at the same time the SLA is guaranteed. What is more, virtualization technology is able to carry out remapping between virtual machine (VM) and physical resources according to the load change so as to achieve the load balance of the whole system in a dynamic manner [3]. Therefore, virtualization technology is being comprehensively used in cloud computing. However, due to the highly dynamic heterogeneity of resources on cloud computing platform, virtual machines must adapt to the cloud computing environment dynamically so as to achieve its best performance by fully using its service and resources. But in order to improve resource utility, resources must be properly allocated and load balancing must be guaranteed [4]. Therefore, how to schedule VM resources to realize load balancing in cloud computing and to improve resource utility becomes an important research point.

Currently in cloud computing, it mainly considers the current system condition in VM resources scheduling but seldom considers the pervious condition before scheduling and the influence on system load after scheduling which usually leads to load imbalance. Most of the load balancing exists in VM migration [5]. Yet, when the entire VM resources are migrated, due to the large granularity of VM resources and the great amount of data transferred in migration and the suspension of VM service, the migration cost becomes a problem. This paper presents a scheduling strategy to realize load balancing. According to historical data and current state and through genetic algorithm, this method computes in advance the influence it will have when the current VM service resources that need deploying are arranged to every physical node, then it chooses the deployment that will have the least influence on the system. In this way, the method realizes the best load balancing and reduces or avoids dynamic migration.

II. RELATED WORK

Load balancing has always been a research subject whose objective is to ensure that every computing resource is distributed efficiently and fairly and in the end improves resource utility. In traditional computing environments of distributed computing, parallel

doi:10.4304/jcp.7.1.42-52

computing and grid computing, researchers in and abroad have proposed a series of static and dynamic and mixed scheduling strategies [6]. In static scheduling algorithm, ISH [7], MCP [8] and ETF [9] algorithms based on BNP are suitable for small distributed environments with high internet speed and ignorable communication delay while MH [10] and DSL [11] algorithm based on APN take into consideration of the communication delay and execution time so they are suitable for larger distributed environments. In dynamic scheduling algorithm, some algorithms guarantee the load balancing and load sharing in task distribution through self-adapting distribution and intelligent distribution. In mixed scheduling algorithm, it mainly emphasizes equal distribution of assigned computing task and reduction of communication cost of distributed computing nodes and at the same time it realizes balanced scheduling according to the computing volume of every node. Researchers have also conducted studies on algorithms of autonomic scheduling, central scheduling, intelligent scheduling and agent negotiated scheduling. There are many similarities and also differences between traditional scheduling algorithms and the scheduling of VM resources in cloud computing environment. First, the biggest difference between cloud computing environment and traditional computing environment is the target of scheduling. In traditional computing environment, it mainly schedules process or task so the granularity is small and the transferred data is small; whereas in cloud computing environment the scheduled target is VM resources so the granularity is large and the transferred data is large as well. Second, in cloud computing environment, compared with the deployment time of VMs, the time of scheduling algorithm can almost be neglected. This paper sees to the equal distribution of hardware resources of VMs in cloud computing environment so that the VM can improve its running efficiency while meeting the QoS needs of subscribers.

At present, a number of studies on the balanced scheduling of VM resources are based on dynamic migration of VMs. Sandpiper [12] system carries out dynamic monitoring and hotspot probing on the utility of system's CPU, Memory resources and network bandwidth. It also puts up with the resource monitoring methods based on black-box and white-box. The focus of this system is how to define hotspot memory and how to dispose hotspots through the remapping of resources in VM migration. VMware Distributed Resource Scheduler (DRS)[13] is a tool to distribute and balance computing volume by using the available resources in virtualized environment. VMware DRS continuously monitors resource utility over the resources pool then conducts intelligent distribution of available resources among several VMs according to the predefined rule which reflects business needs and the changing priority. If there is dramatic change of workload in one or more VMs, VMware DRS will redistribute VMs among physical servers and migrate VMs to different physical servers through VMware VMotion. All of the above systems achieve system load balance through dynamic migration, but frequent dynamic migration will employ a large number of resources which finally leads to performance degrading of the whole system.

Though, there are many opensource cloud systems for researchers emerges as the development of cloud computing. For instance, there are some popular open-source cloud systems, such as Eucalyptus [14], Open Nebula [15], Nimbus [16], etc. To decide the allocation, Eucalyptus uses Greedy (First fit) and Rotating algorithm [14], Open Nebula uses queuing system, advanced reservation and preemption scheduling [15], and Nimbus uses some customizable tools like PBS and SGE [16]. In the above scheduling approaches, Greedy and Rotating that provided by Eucalyptus is a random method to select adaptive physical resources for the VM requests that not considering maximum usage of physical resource. The queuing system, advanced reservation and preemption scheduling policies are not considering the utilization rate of physical resource. For customizable strategies, are basic queuing systems that do not provide automated optimal resource scheduling and being indeterminate.

Genetic algorithm [17] is a random searching method developed from the evolution rule in ecological world (the genetic mechanism of survival of the fittest).It has internal implicit parallelism and better optimization ability. By the optimization method of probability, it can automatically obtain and instruct the optimized searching space and adjust the searching direction by itself. Considering the VM resources scheduling in cloud computing environment and with the advantage of genetic algorithm, this paper presents a balanced scheduling strategy of VM resources based on genetic algorithm[18][19][20][21]. According to historical data and current states, this method computes in advance the influence it will have when the current VM service resources that need deploying are arranged to every physic node, based on which the method achieves the best load balancing. In the first part of this paper, it introduces the current situation of VM resources scheduling in cloud computing environment; in the second part, it designs the VM scheduling model; in the third part, it raises the VM resources scheduling method based on genetic algorithm; at the end, an analysis of the method is made and an experiment and summary is also conducted.

III. SCHEDULING ARCHITECTURE IN CLOUD COMPUTING ENVIROMENT

According to the popular cloud systems, the computational resources are usually connected by LAN. The cloud is somehow centralized and we just need to consider the "scheduler" [22]. Figure 1 illustrates the popular standard-based cloud architecture, and the scheduler is always at the top lawyer [23].

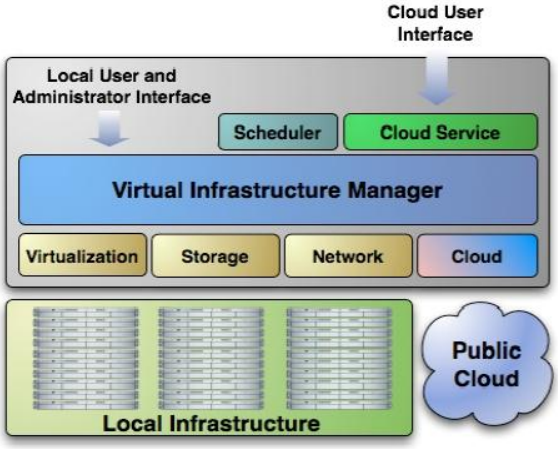


Figure 1. Standard-based cloud virtual infrastructure manager

Figure 2 shows the mapping relationship between VMs and physical machines. Cloud provides all kinds of machines it possesses in forms of virtual machine that clients can visit it through Internet as a service, and it always play a role of scheduling sever. And the computing nodes are different kinds of ordinary PCs, servers, and even high performance clusters in which we will set up VMs.

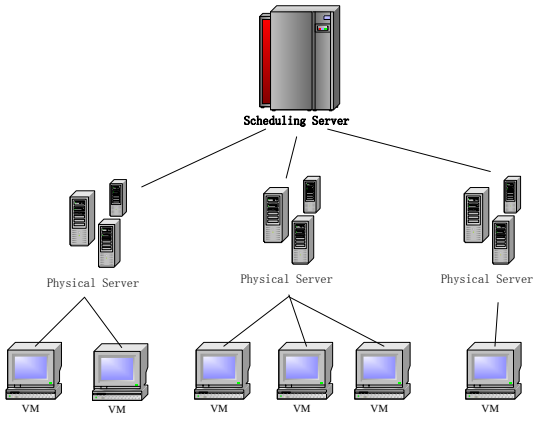


Figure 2. Scheduling Architecture in Cloud Computing Environment

There are many scheduling strategies in the popular cloud systems. Eucalyptus uses Greedy (First fit) and rotating scheduling strategies. Greedy query all the computational resources from the first to the last node until finding a suitable node every time new request comes and deal with them one by one for multiple requests. Rotating records the last position of the scheduler visited. And the scheduler starts from the last visited position next time new request(s) come(s) meanwhile the resources are considered as a circular linked list. OpenNebula uses Haizea [24], an opensource VM-based lease management architecture as the scheduler and provides the queuing system, advanced reservation, preemption, immediate lease strategies, etc. All these policies pay more attention to "when" but neglect "how". Nimbus can be configured to use familiar schedulers like PBS (Portable Batch System) or SGE (Sun Grid Engine) to schedule virtual machines [25][26][27]. PBS is a queuing system and SGE uses Job Scheduling Hierarchically (JOSH), both do not have a good utilization of resources.

IV. THE MODEL DESIGN OF VM SCHEDULING

A. VM Model

From Figure 2, we can see the mapping relationship between VMs and physical machines. The set of all the physical machines in the system is $P=\{P_1,P_2,\cdots,P_N\}$, N is the number of physical machines, $P_i(1\le i\le N)$ stands for physical machine No.i. We name the VMs set on physical machine P_i $V_i=\{V_{i1},V_{i2},,V_{im_i}\}$ in which m_i is the number of VMs on physical machine No.i. Suppose we need to deploy VM V at present, and we use $S=\{S_1,S_2,\cdots,S_N\}$ to represent the mapping solution set after V is arranged to every physical machine. S_i here refers to the mapping solution when VM V is arranged to physical machine P_i.

B. The Expression of Load

The load of a physical machine usually can be obtained by adding the loads of the VMs running on it. We suppose the best time span monitored by historical data is T. That is, the time zone of T from the current time is the monitoring zone by historical data. According to the varying law of physical machine load, we can divide time T into n time periods. Thus we hereby define $T=[(t_1-t_0),(t_2-t_1),\cdots,(t_n-t_{n-1})]$. In the definition, (t_k-t_{k-1}) refers to time period k. Suppose the load of VMs is relatively stable in every period, then we can define the load of VM No.i in period k is $V(i,k)$. Therefore, we can conclude that in cycle T, the average load of VM V_i on physical machine P_i is

$$\overline{V_i(i,T)}=\frac{1}{T}\sum_{k=1}^{n}V(i,k)\times(t_k-t_{k-1}) \tag{1}$$

According to the system structure, the load of a physical machine usually can be obtained by adding the loads of the VMs running on it. Therefore we can conclude the load of physical machine P_i is

$$P(i,T)=\sum_{j=1}^{m_i}\overline{V_i(j,T)} \tag{2}$$

The current virtual machine needs deploying is V. Since the resources information needed by the current deployment VM has already been defined, we can estimate the load of the VM is V' based on relevant information. So when VM V is arranged to physical machine, the load of every physical machine should be

$$P(i,T)'=\begin{cases}P(i,T)+V' & \text{After Deploy } V\\ P(i,T) & \text{Others}\end{cases} \tag{3}$$

Usually, when VM V is arranged to physical machine P_i, there will be a certain change in system load. Thus

we need to carry out load adjustment to achieve load balancing. The load variation of mapping solution S_i in time period T after VM V is arranged to physical machine P_i is

$$\sigma_i(T) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\overline{P(T)'} - P(i,T)')^2} \quad (4)$$

where

$$\overline{P(T)'} = \frac{1}{N}\sum_{i=1}^{N} P(i,T)' \quad (5)$$

C. Mathematical Model

Through the previous analysis, we define the following mathematical model:

Definition 1: Under system mapping solution S_i, the load of every physical machine is $P(i,T)'$, and the total load variation (mean square deviation to the average load) in time period T is defined as

$$\sigma_i(S_i,T) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\overline{P(T)'} - P(i,T)')^2} \quad (6)$$

where

$$\overline{P(T)'} = \frac{1}{N}\sum_{i=1}^{N} P(i,T)' \quad (7)$$

Definition 2: the balanced mapping solution of system mapping solution S_i is S_i', and then the set of mapping solution S should correspond to the set of balanced mapping solution $S' = \{S_1', S_2', \cdots, S_N'\}$. S_i' is the best mapping solution to make $\sigma_i(S_i,T)$ meet the predefined load constraints.

Definition 3: we define the ratio of VM number M' need migrating to achieve load balancing in a certain mapping solution to the total VM number M as cost divisor. Then for every mapping solution S_i, the cost divisor $\rho(S_i)$ to reach load balancing S_i' is defined as

$$\rho(S_i) = \frac{M'}{M} \quad (8)$$

The objective of this paper is to find the best mapping solution S_i so as to achieve the best system load balancing or rather, to minimize the cost divisor $\rho(S_i)$ in load balancing. We can obtain the best mapping solution S_i' from mapping solution S_i though genetic algorithm.

V. REALIZATION OF BALANCED SCHEDULING THROUGH GENETIC ALGORITHM

Genetic algorithm is a random searching method developed from the evolution law in the ecological world. After the first population is produced, it evolves better and better approximate solutions based on the law of survival of the fittest and from generation to generation. In every generation, the individual is chosen based on the fitness of different individuals in a certain problem domain. Then the individuals combine and cross and vary by the genetic operators in natural genetics and then a new population representing a new solution set is produced. Based on the real situation of cloud computing, this paper presents a scheduling strategy through genetic algorithm.

A. Population Coding

To tackle problems by genetic algorithm, it is not to function on the solution pool but to produce a certain coding denotation. So first we need to do the coding for the problem to be tackled. The selection of the coding method to a great extent depends on the property of the problem and the design of genetic operators. The classic genetic algorithm marks the chromosome structure of genes by binary codes. Judged from the data model in this paper, it can be found that it is a one-to-many mapping relationship between physical machines and VMs. Therefore, this paper chooses tree structure to mark the chromosome of genes [28]. That is to say, every mapping solution is marked as one tree; the scheduling and managing node of the system on the first level are the root nodes while all of the N nodes on the second level stand for physical machines and the M nodes on the third level stand for the VMs on a certain physical machines.

B. Initialization of Population

For the initialization of population, this paper mainly uses the method of spanning tree. We have the following definitions for the tree:

- This tree is a spanning tree constructed by the elements in the physical machine set and VM set.
- The root node of this tree is the predefined management source node.
- All of the physical machine nodes and VM nodes are included in this tree.
- All of the leaf nodes are VM nodes.

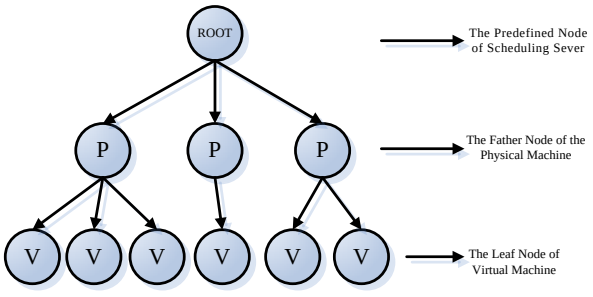


Figure 3. The spanning tree of the initialized population

The principle of the spanning tree is that it should meet the given load balancing conditions or it should produce relatively fine descendents through inheritance. This means the tree itself should also be a comparatively fine individual. Therefore we can get the mapping relationship between physical machines and VMs through the following procedures. First, we compute the selection

probability p (p is the ratio of a single VM load to the load sum of all the VMs) of every VM according to the VM load in the VM set; then based on the probability p all of the VMs are allocated to the smallest-loaded node in the physical machine set to produce the leaf node of the initial spanning tree. In this way, the possibility of those VM with more heat being selected is raised and those VM with low heat can also be selected.

C. Fitness Function

In the natural world, an individual's fitness is its productivity which directly relates to the number of its descendents. In genetic algorithm, fitness function is the criterion for the quality of the individuals in the population. It directly reflects the performance of the individuals – the better the performance, the bigger the fitness, vice versa. The individuals are decided to multiply or to extinct by the value of the fitness function. Therefore, fitness function is the driving force of genetic algorithm. The fitness function in this paper is

$$f(S,T)=\frac{1}{A+B\times f_H} \tag{9}$$

$$f_H=\Phi(\sigma_i(S,T)-\sigma_0),\Phi(X)=\begin{cases}1, X\le 0\\ r, X>0\end{cases} \tag{10}$$

Where, A and B are weighted coefficients which are defined in concrete application. σ_0 stands for the heat variation constraints permitted in system load balancing and can be predefined. $\Phi(X)$ is penalty function in which the value is 1 when the individual meets the correspondent constraints; otherwise the value is r which can also be defined according to concrete situations.

D. Selection Strategy

Selection strategy means to select the individual of next generation according to the principle of survival of the fitness. Selection strategy is the guiding factor for genetic performance. Different selection strategies will lead to different selection pressure or rather, the different distribution relationship of parental individuals of next generation. The algorithm in this paper mainly uses the selection strategy based on fitness ratio.

First we work out the fitness of the individuals in current population by fitness function, and we keep the individual with the highest fitness into the child population; then we compute the selection probability of the individuals according to their fitness values.

$$p_i(S)=\frac{f_i(S,T)}{\sum_{i=1}^{D} f_i(S,T)} \tag{11}$$

Where, $f_i(S,T)$ stands for the fitness of member No.i in the population; D stands for the scale of the population.

Lastly, we conduct election of the individuals by the rotating selection strategy so that the individual with the high fitness has higher probability being selected and those with low fitness also have the chance to be chosen.

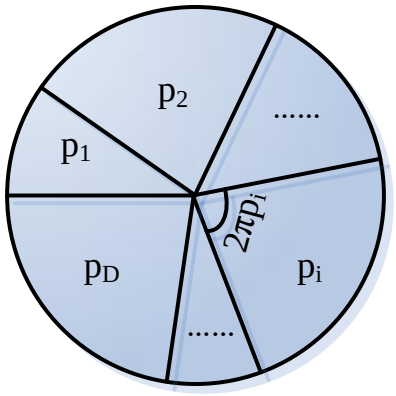


Figure 4. The circle of the rotating selected strategy

The rotating selection strategy according to the selection probability $p_i (i = 1,2,\ldots,D)$ and based on the population scale divides a circle into D parts, among which the central angle of No.i is $2\pi p_i$ as is shown in Figure 4. Spin the circle until it stops. If some reference point stops within the sphere of No.i, then select the No.i individuals. To realize this, we need to get a random number k, $k\in[0, 1]$, whereas if $p_1 + p_2 + \ldots + p_{i-1} < k \le p_1 + p_2 + \ldots + p_i$, then choose the No.i individuals. In this way, the bigger the fitness value, the bigger the area it takes in the sector, and the bigger chance of being selected.

E. Crossover Operation

Crossover operation is to produce new individuals by substituting and reforming parts of the two subsequently selected parental individuals. Through hybridization the searching ability of genetic algorithm gets tremendous improvement. Since genetic algorithm uses tree coding, so in order to ensure the validity of the chromosome of the descendents, the algorithm here cannot do the hybridization like the genetic algorithm using binary coding which simply exchanges parts of the genes[28]. This paper simulates the hybridizing process of life-beings to ensure the descendents intake the same gene from the parental chromosome and also to guarantee the validity of the trees of the descendents. The hybridization operators are as Figure 5.

- Choose two parental individuals T_1 and T_2 according to the rotating selection algorithm;
- Combine the two parental individuals to form a new individual tree T_0 which keeps the individuals with the same leaf nodes in the two parental individuals and disposes the different ones;
- For the different leaf nodes in the two parental individuals, first compute their selection probability p according to the load of every VM, then based on p distribute them as leaf nodes to the smallest-loaded nodes in the physical machine set until the distribution is completed;
- Repeat the above procedures until the produced individuals reach the number required.

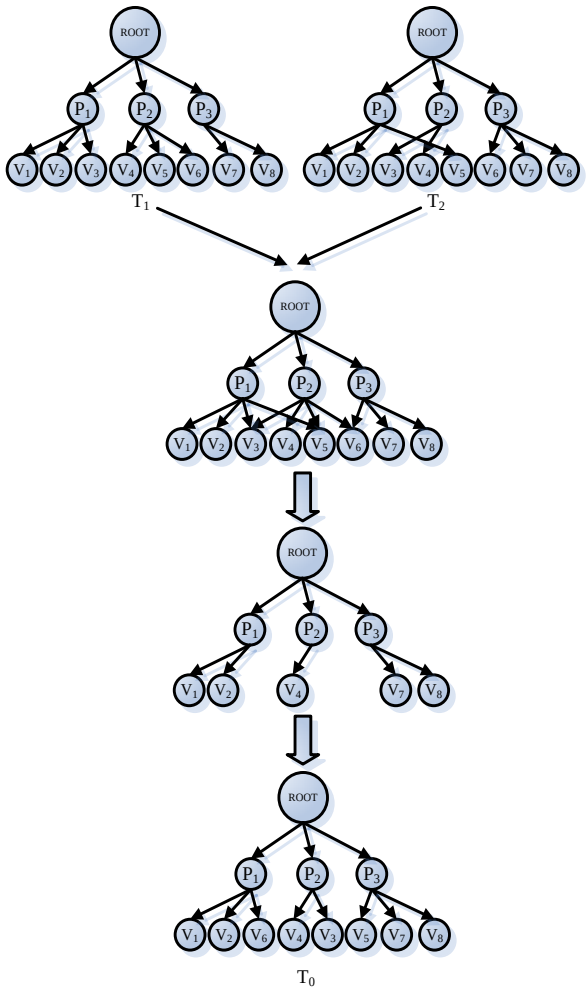


Figure 5. Crossover operation

F. Mutation Operation

In order to get bigger variation operators in the beginning of genetic operation to maintain the variety of the population and avoid prematurity, the variation operator is reduced to ensure the regional searching ability when the algorithm gets close to the best solution vicinity. This paper uses the following self-adaptive variation probability.

$$P_m = \exp(-1.5 \times 0.5t) / D \times \sqrt{M} \qquad (12)$$

Where, t is the number of generations; D is the scale of the population; and M is the number of VMs.

The individuals are randomly chosen to vary according to the variation probability. Besides, to avoid the reoccurrence of the same gene on the same one chromosome, when the gene on one locus varies in this chromosome, the gene on the correspondent locus of the varied gene code should consequently change into the original gene code of the varied locus. That is to say, the leaf nodes should be changed after variation.

G. Scheduling Strategy

The objective of this paper is to find the best mapping solution to meet the system load balance to the greatest extent or to make the cost gene of load balancing the lowest. We want to find the best scheduling solution for the current scheduling through genetic algorithm. And the terminating condition of this hunting for the best scheduling solution is the existence of a tree that meets the heat restriction requirement. We first compute the cost gene through the ratio of the current scheduling solution to the best scheduling solution, and then we decide the scheduling strategy according to the cost gene. We choose the scheduling solution with the lowest cost as the final scheduling solution so that it has the least influence on the load of the system after scheduling and it has the lowest cost to reach load balancing. In this way, the best strategy is formed.

VI. ALGORITHM ANALYSIS

A. Global Scheduling Algorithm

Considering the VM resource scheduling in cloud computing environment and with the advantage of genetic algorithm, this paper presents a balanced scheduling strategy of VM resources based on genetic algorithm. Starting from the initialization in cloud computing environment, we look for the best scheduling solution by genetic algorithm in every scheduling. When there are no VM resources in the whole system, we use the algorithm to choose the scheduling solution according to the computed probability; with the increase of VM resources and the increase of running time, according to historical data and the current state we compute in advance the influence it will have when the current VM service resources that need deploying are arranged to every physic node, and then choose the best solution. The main procedures are as follow.

Step 1: In initialization, there are not any VM resources in the system so there is no historical information. When there are VM resources to be scheduled, based on the computed probability, the algorithm randomly chooses the free physical machine and starts scheduling;

Step 2: With the increase of VM resources in the system and the increase of running time, according to historical information and the current state, the algorithm computes the load and variance of every physical machine in every solution from the scheduling solution set S.

Step 3: The algorithm uses genetic algorithm to compute the best mapping solution for every solution in S. The best solution refers to the one in which the variance meets the predefined load constraints;

Step 4: The algorithm computes respectively the costs or cost divisors of every solution in S to achieve the best mapping solution;

Step 5: According to the cost divisor of every solution, the algorithm chooses the one with the lowest cost as the scheduling solution and completes the scheduling;

Step 6: Should there be new VM resources need scheduling, then go back to step 2.

In every scheduling, we use genetic algorithm to find the best scheduling solution; and in the next scheduling, because of the accumulation of the best solutions by the original scheduling solutions, the best scheduling

solution can always be found to achieve load balancing. Even though there is big load variation in the system due to special reasons and one time scheduling cannot achieve system load balance, the method can still find a scheduling solution with the lowest cost to achieve load balancing of the system.

B. Astringency Analysis of the Genetic Algorithm

To test the astringency of the genetic algorithm, we carry out the following experiment. We suppose the number of physical machines is 5 and the number of started VMs is 15. The mapping relationship between physical machines and VMs is shown in Figure 6. The average load of every VM in period T is shown in Table 1. Meanwhile according to the whole system condition, we make the following supposition. The scale of the population D=50, replication probability $P_r = 0.1$, hybridization probability $P_c = 0.9$, variation probability is self-adaptive probability. Besides, according to the theory, we conclude that hybridization probability $P_c \in [0,1]$, variation probability $P_m \in (0,1)$. When the system load variation constrain $\sigma = 0.5$, through the experiment we finally attain a new mapping solution shown in Figure 6.

TABLE I. VM AVERAGE LOAD

Virtual Machine	CPU Utility	Virtual Machine	CPU Utility
V1	28.8	V9	18.0
V2	23.4	V10	9.2
V3	17.9	V11	8.8
V4	16.8	V12	7.3
V5	12.6	V13	8.1
V6	22.3	V14	28.8
V7	13.9	V15	24.0
V8	40.2	V16	26.9

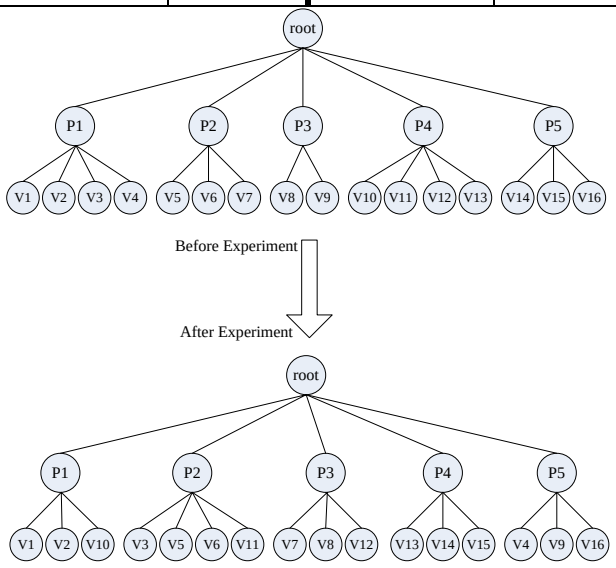


Figure 6. Mapping relationship before and after using the algorithm

Through the experiment, we get the mapping relationships before and after using the algorithm respectively. The results are shown in Figure 6. It can be seen in the figure that after using the algorithm, the loads of every node basically tend to be balanced and the system load variation is smaller than σ. Therefore we can conclude that the algorithm has fairly good global astringency and can converge to the best solution in a very short time.

C. Efficiency Analysis of the Genetic Algorithm

This paper through selecting different number of physical machines and virtual machines and making a larger number of experiments attain CPU execution time of the best solution under different number of VMs. From the Figure 7 shown we can see that with the increase of VM number, there is no significant increase of execution time for this genetic algorithm and it can still keep a good performance, which proves that the efficiency of this genetic algorithm is relatively high.

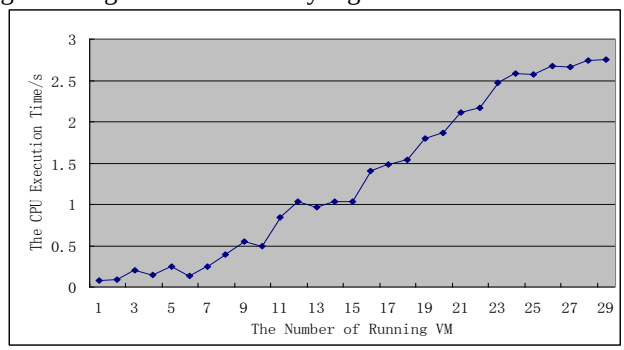


Figure 7. CPU execution time under different number of VMs

VII. EXPERIMENT AND RESULTS ANALYSIS

After the above verification of the astringency of the genetic algorithm, in order to further assess the performance of the global algorithm, we carried out the experiment on the Platform ISF® [29] and open-source VM management platform OpenNebula. We chose a physical machine as the host machine in which we installed front-end to manage and schedule VM; and its operation system is RHEL5.4, the CPU is Intel® Core™ 2 Duo 3.0GHz, and the Memory is 2.0GB. Meanwhile, we chose 20 physical machines as client machines in which we installed Agent client and KVM VM; and the operation system is Ubuntu 10.04, CPU is Intel® Core™ 2 Duo 3.0GHz and Memory is 2.0GB, and the disk capacity is 320GB. The whole network was connected by LAN (Local Area Network). In the experiment, the host machine was the root node; the client physical machines were the second level nodes and the VM client operation systems on the physical machines were child nodes. The whole algorithm was realized by C++.

To better test the stability of the algorithm, we define VM load variation rate as α which indicates the variation range of VM load. Suppose the initial VM load deployed is $L(T_0)$, the current VM load is L(t), then the load variation rate is:

$$\alpha = \frac{|L(T_0) - L(t)|}{L(T_0)} \tag{13}$$

The experiment mainly analyzes the load balancing

effect of the algorithm and the migration cost to realize the system load balancing after scheduling by the algorithm, and makes relevant comparisons between this algorithm and the current VM balancing scheduling methods including the least-loaded scheduling method and the rotating scheduling method.

A. *Algorithm Effect Analysis*

Usually in real application environment, the instant load of every physical machine cannot reflect the real load situation. When there are a large number of users for the VM resources of a certain server, the average load of the sever in this period of time will be relatively high; whereas at some moment when there is a big loss of users or a crush of users which makes the instant load of the server too low or too high, then this load value is not the real reflection of the server. The least-load method chooses the server with the lowest load to schedule based on the instant load of every current server. Therefore, when the resource utilization of the server is relatively stable within a certain time and the variation of the server's load is relatively subtle, the scheduling will be relatively balanced; however, if the resource utilization of the server is variant in a certain period of time, the instant load value will not be able to reflect the load situation of the server because the variation of the server's load is too big, and if the instant load value happens to be that of the wave crest or the wave trough, the scheduling under the least-load method will be severely affected. As for the rotation scheduling method, it first numbers the physical machines and then chooses the next physical machine to schedule without considering the load situation of every physical machine. Thus the system load will be highly unbalanced while the load variation is big. In the scheduling algorithm of this paper, the load value of every physical machine is able to reflect the real load situation of the server thanks to its consideration of the comprehensive load situation of every physical machine within a certain period of time. According to historical data and current state and through genetic algorithm, this method computes in advance the influence it will have when the current VM service resources that need deploying are arranged to every physical node, then it chooses the deployment that will have the least influence on the system. In this way, the load balance of the system can be well kept after scheduling both when the system load is stable and variant.

We deploy one virtual machine every other ten seconds to the system. All together we deploy 100 virtual machines, each of whose initial resource utility ranges from 5% to 30%. We randomly choose five physic machines and observe their load situation. When system load variation is relatively small, that is, when $\alpha<0.1$, the load trend variation figure of rotating scheduling algorithm, least-load scheduling algorithm and the algorithm of this paper are shown in figure 8. This experiment mainly analyzes the load balancing effect of the algorithm, and compares this method with the least-loaded scheduling method and the rotating scheduling method in two different situations. It can be seen from the figure that on many occasions the five curves in the algorithm of this paper almost overlap which indicates that this algorithm has good load balancing effect. However, the five curves in the least-load algorithm divert to a great extent from each other which indicates that the load balancing effect is not as good as the algorithm of this paper; the bigger diversion of the five curves in rotating scheduling algorithm indicates a worse balancing effect of the rotating algorithm. But on the whole, when the system load is comparatively stable, that is, the system load variation is relatively small ($\alpha<0.1$), all of the three methods are able to ensure the system load balancing to a certain extent; while when the system load variation is evident, that is, the system load variation is relatively large, whether the algorithm of this paper can better guarantees the system load balancing will be discussed later.

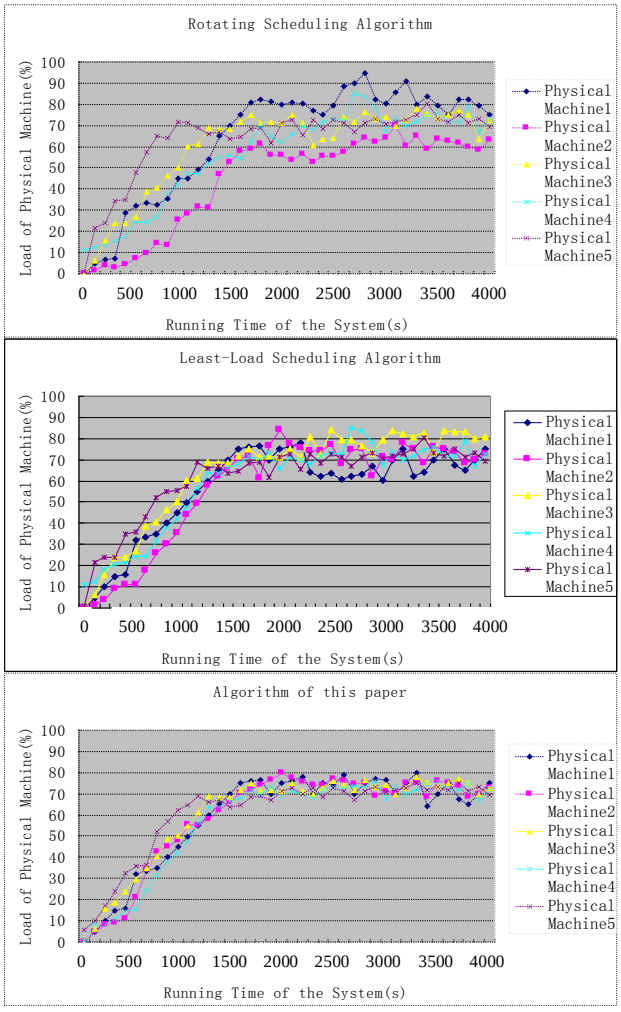


Figure 8. Comparison of three algorithms when the system load variation is relatively small ($\alpha<0.1$)

B. *Influence of Vibration Rate on Load Balancing*

The load vibration rate of virtual machines has significant influence on load balancing[30], therefore it is necessary to make a study on the relationship between variation rate α and system load balancing.

To measure the effect of load balancing, we introduce the load distance LD to mark the distance among the five

curves. At any moment, LD refers to the added distance between every two points on the five curves. The closer the five curves, the smaller LD is and the more balanced the system load is.

Suppose G is the group of all load observation points, N is the total number of observation points. If P is one certain point, then the load of five servers observed at point P are L_0, L_1, L_2, L_3, L4. And the load distance LD_P at point P is:

$$LD_P = \sum_{i=0}^{3} \left(\sum_{j=i+1}^{4} |L_i - L_j| \right) \quad (14)$$

We measure the whole load balancing effect by the average load distance which refers to the mean of the load distance at all observation points. The computing formula of this mean is:

$$\overline{LD} = \frac{\sum_{P \in G} LD_P}{N} \quad (15)$$

In order to examine the influence of VM load variation rate α on average load distance, we work out the correspondent average load distance respectively when α is 0.1, 0.2, …, 0.9. The result is shown in figure 9. It indicates that load variation has very little influence on the load balancing effect of the algorithm in this paper; in contrast, it has bigger influence on least-load algorithm and the biggest influence on rotating scheduling algorithm.

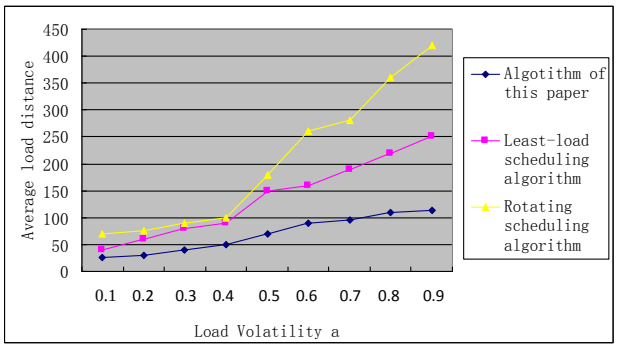


Figure 9. Comparison of three algorithms when the system load variation is relatively changed

C. Migration Cost Analysis

In cloud-computing environment, the resources of a specific VM provide specific service, such as the specific software resources and computing resources. However, the uncertainty of users usually leads to the uncertainty of the utilization of server resources. Consequently there will be big variation in the load of every physical machine. And very often the system still needs a dynamic migration in a period of time to realize load balance after the scheduling. The scheduling effects of the least-load method and the method in this paper are relatively satisfactory when then resources utilization is certain or rather the variation of the system load is small; and the migration cost caused by the system variation after scheduling is low; in contrast, due to its ignorance of the load situation of the system. The rotation scheduling method brings high migration cost after scheduling both when the system load is stable and variant. What is more, since the least-load method only takes into consideration of the instant load of very server during system scheduling which is not able to reflect the real load of physical servers, it will probably lead to the serious unbalance for the system load after scheduling. The rotation scheduling method may also lead to the increase of the system migration cost which means the VM number will be increased to achieve system load balancing and the migration cost of the algorithm will be raised sharply with the increase of the system load variation. The method in this paper according to historical data and current state and through genetic algorithm, computes in advance the influence it will have when the current VM service resources that need deploying are arranged to every physical node, then chooses the deployment that will have the least influence on the system. In this way, the problem of load imbalance of the system after scheduling can be avoided to a great extent, and the migration cost after scheduling is reduced to the lowest level.

On some special occasions, there is a big increase of the load of some nodes in the system due to frequent access thus leads to the load imbalance of the whole system. Under this situation, usually the system cannot realize the system load balancing through only one-time scheduling so it must do it through VM migration. However, the cost of VM migration cannot be neglected. Thus where the VM should be migrated and how to migrate the least number of VM are also the problems that need consideration during VM scheduling. The algorithm of this paper takes historical factors into consideration. It computes the situation of the whole system after scheduling in advance through genetic algorithm and then chooses the scheduling solution with the lowest cost. Figure 10 shows the average VM migration ratio while the VM load variation rate α is changing. It can be seen that the method of this paper shows conspicuous advantage. The experiment shows that the method of this paper can greatly brings down the migration cost.

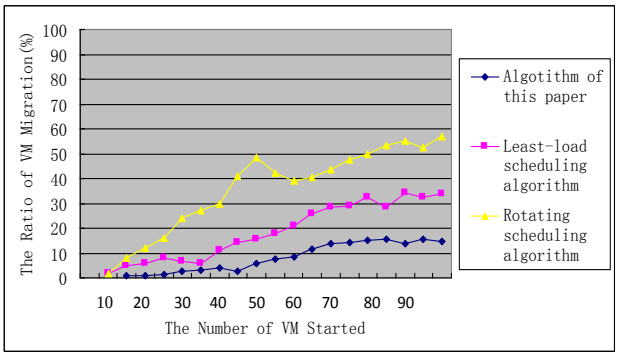


Figure 10. The migration ratio of VM when the number of started VM is different

D. Utilization Rate of the Algorithm

We investigate the utilization rate of the algorithm in this paper, Least-loaded, Rotating algorithm and the average utilization for other queuing and configurable scheduling. From the Figure 11, we can work out how much resource each model wasted when allocating different number of VMs. And we find that sometimes

the Least-loaded, Rotating and queuing systems cannot allocate resources for all the VMs even if there are enough resources for the VMs. But the algorithm in this paper always gives a good scheduling as long as there are enough resources. On the other hand, we can see that the algorithm in this paper saves the most resources.

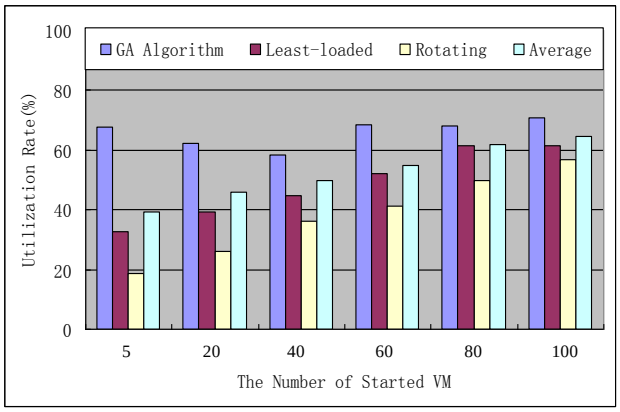


Figure 11. Comparison of three algorithms for the utilization rate

VIII. CONCLUSIONS

In view of the current load balancing in VM resources scheduling, this paper presents a scheduling strategy on VM load balancing based on genetic algorithm. Considering the VM resources scheduling in cloud computing environment and with the advantage of genetic algorithm, this method according to historical data and current states computes in advance the influence it will have on the whole system when the current VM service resources that need deploying are arranged to every physical node, and then it chooses the solution which will have the least influence on the system after arrangement. In this way, the method achieves the best load balancing and reduces or avoids dynamic migration thus resolves the problem of load unbalancing and high migration cost caused by traditional scheduling algorithms. The experimental results show that this method can better realize load balancing and proper resource utilization.

This paper builds a model based on the concrete situations of cloud computing. It considers the historical data and current states of VM, uses tree structure to do the coding in genetic algorithm, proposes the correspondent strategies of selection, hybridization and variation also puts some control on the method so that it has better astringency. However in real cloud computing environment, there might be dynamic change in VMs, and there also might be an increase of computing cost of virtualization software and some unpredicted load wastage with the increase of VM number started on every physical machine. Therefore, a monitoring and analyzing mechanism is needed to better solve the problem of load balancing. This is also a further research subject.

ACKNOWLEDGMENT

This work is supported by The National "863" Program of China (Grant No.2009AA01Z142). Thanks for the support from Platform Corporation and High Performance Computing center of Northwestern Polytechnical University.

REFERENCES

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, “Above the clouds: A berkeley view of cloud computing,” UC Berkeley Technical Report UCB/EECS-2009-28, February 2009.

[2] Borja Sotomayor, Kate Keahey, and Ian Foster, “Overhead matters: A model for virtual resource management,” In VTDC '06: Proceedings of the 1st International Workshop on Virtualization Technology in Distributed Computing, page 5, Washington, DC, USA, 2006.

[3] Borja Sotomayor, Kate Keahey, Ian Foster, and Tim Freeman, “Enabling cost-effective resource leases with virtual machines,” In Hot Topics session in ACM/IEEE International Symposium on High Performance Distributed Computing 2007 (HPDC 2007), 2007.

[4] L. Cherkasova, D. Gupta, and A. Vahdat, “When virtual is harder than real: Resource allocation challenges in virtual machine based it environments,” Technical Report HPL-2007-25, February 2007.

[5] Clark C, Fraser K, Hand S, “Live Migration of Virtual Machines[C],” Proceedings of the 2nd Int’l Conference on Networked Systems Design & Implementation. Berkeley, CA, USA, 2005.

[6] Wei Wang, “A reliable dynamic scheduling algorithm based on Bayes trust model,” Computer Science, 2007.

[7] Rewinin H E, Lewis T G, Ali H H, “Task Scheduling in parallel and Distributed System Englewood Cliffs,” New Jersey: Prentice Hall, 1994, pp. 401-403.

[8] Wu M, Gajski D, Hypertool, “A programming aid for message passing system,” IEEE Trans Parallel Distrib Syst, 1990, pp. 330-343.

[9] Hwang J J, Chow Y C, Anger F D, “Scheduling precedence graphics in systems with inter-processor communication times,” SIAM J Comput, 1989, pp. 244-257.

[10] Rewinin H E, Lewis T G, “Scheduling parallel programs onto arbitrary target machines,” J Parallel Distrib Comput, 1990, pp. 138-53.

[11] Sih G C, Lee E A, “A compile-time scheduling heuristic for Interconnection-constraint heterogeneous processor architectures,” IEEE Trans Parallel Distrib Syst, 1993, pp. 175-187.

[12] Wood T, “Black-box and Gray-box Strategies for Virtual Machine Migration[C],” Proceedings of the 4th Int’l Conference on Networked Systems Design & Implementation, IEEE Press, 2007.

[13] VMWare, “VMware DRS - Dynamic Scheduling of System Resources,” http://www.vmware.com/cn/products/vi/vc/drs.html, 2009.

[14] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. So-man, L. Youseff, and D. Zagorodnov, “ The Eucalyptus open-source cloud-computing system”, IEEE International Symposium on Cluster Computing and the Grid (CCGrid ’09), 2009.

[15] OpenNebula, “OpenNebula Software,” http://www.opennebula.org, 2010.

[16] Nimbus, http://nimbusproject.org, 2010.

[17] E. Goldberg, “The existential pleasures of genetic algorithms,” In: Genetic Algorithms in Engineering and Computer Science, Winter G ed. New York: Wiley, 1995, pp. 23-31.

[18] Kim, H. Kim, M. Jeon, E. Seo, J. Lee, “Guest-aware prioritybased virtual machine scheduling for highly consolidated server,” In Proc. Euro-Par, 2008.

[19] Ongaro, A. L. Cox, and S. Rixner, “Scheduling I/O in virtual machine monitors,” In Proc. VEE, 2008.

[20] L. Cherkasova, D. Gupta, and A. Vahdat, “Comparison of the three CPU schedulers in Xen,” SIGMETRICS Perform. Eval. Rev., 2007, pp. 42–51.

[21] Hu Jinhua, Gu Jianhua, Sun Guofei, Zhao Tianhai. A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment, PAAP2010, 2010, pp. 89-96.

[22] Hai Zhong, Kun Tao, Xuejie Zhang, "An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems", The Fifth Annual ChinaGrid Conference, 2010.

[23] T. Tan and C. Kiddle, "An Assessment of Eucalyptus Version 1.4", Technical Report 2009-928-07, Department of Computer Science, University of Calgary, 2009.

[24] Haizea, http://haizea.cs.uchicago.edu, 2010.

[25] Amazon Web Services, http://aws.amazon.com, 2010.

[26] openPBS, http://pbsgridworks.com, 2010.

[27] SunGridEngine, http://gridengine.sunsource.net, 2010.

[28] Yunzhu Ni, Guanghong Lv, Yanhui Huang, "The Solution of Disk Load Balancing Based on Disk Striping with Geneti Algorithm," Chinese Journal of Computers, 2006.

[29] Platform, "Platform ISF Software", http://www.platform.com.cn/, 2010.

[30] Li WZ, Guo S, Xu P, Lu SL, Chen DX. An adaptive load balancing algorithm for service composition. Journal ofSoftware, 2006, 17(5): 1068—1077.

BIOGRAPHIES

Jianhua Gu, Male, Doctor, Professor, Doctor Supervisor, Specialized in Distributed Computing, High Performance Computing

Jinhua Hu, Male, Master, Specialized in Distributed Computing

Tianhai Zhao, Male, Doctor, Instructor, Specialized in Distributed Computing, High Performance Computing

Guofei Sun, Male, Master, Specialized in Distributed Computing

Task Scheduling Based On Thread Essence and Resource Limitations

Tomer Y. Morad, Avinoam Kolodny, and Uri C. Weiser
Department of Electrical Engineering, Technion, Haifa, Israel
Email: {tomerm@tx, kolodny@ee, uri.weiser@ee}.technion.ac.il

Abstract—**Scheduling of threads based on the concept of thread essence is proposed in this paper. Multithreaded applications contain serial phases (single thread) and parallel phases (many threads). We propose a thread assignment mechanism that takes into account the essence of the threads in simultaneously-running applications that grants higher priority to applications during their critical-serial phases, for environments where there are more threads than cores. Furthermore, our proposed scheduler considers the limited resources of the system by reducing the number of context switches when there are more ready threads than cores. Analytic and experimental evaluation of the proposed thread assignment mechanism on both symmetric and emulated asymmetric multiprocessors show throughput improvements by as much as 16%, improved fairness by as much as 26% and reduced jitter by as much as 88%.**

Index Terms—**Asymmetric Multiprocessors, Operating Systems, Scheduling**

I. INTRODUCTION

Operating system schedulers' most important task is to allocate the available system resources to the workload defined by the users. Workloads consist of a set of threads, each with its own requirements, such as memory bandwidth, IO, floating point unit usage, branch prediction, and dependency on other threads. The resources of the system are comprised of the available processing cores, their type and their performance, the memory hierarchy, volatile and non-volatile memory, network, graphics and more.

The schedulers carry out the scheduling task by assigning resources to threads, and in most cases also revoking resources from threads, as done by operating systems that support preemption. There are various metrics for assessment of a scheduler's performance, such as system throughput, latency, fairness, jitter, and power consumption.

Schedulers must consider the different attributes of threads in order to achieve good scheduling performance. Threads may have user-defined attributes, such as static priorities, target performance, and maximum allowed power consumption. Another class of attributes includes platform-independent attributes, such as the memory footprint, shared data with other threads, types of instructions, memory access pattern, IO access pattern and more. A third class of attributes includes platform-dependent attributes, such as branch prediction confidence level, miss rate, instructions per cycle (IPC) and more. All these attributes of threads are referred to in this paper as the essence of threads.

An example of a platform independent attribute is the "interactivity" metric used by the Linux scheduler [1]. The Linux scheduler classifies threads as interactive, or IO-bound, when their sleep time, which is the time threads voluntarily release the processor, is high. IO-bound threads are granted higher priority by the Linux scheduler, since these threads exhibit long waits between relatively short CPU usage times. Granting priority to IO-bound threads results in lower latency and better performance perceived by the users.

Indeed, there are papers that suggest considering some of the thread attributes when scheduling threads. Knauerhase et al. [13] suggest co-scheduling threads based on cache usage. Zhuravlev et al. [30] suggest co-scheduling threads based on memory bandwidth usage, memory controller contention and contention on the prefetching hardware.

One of the attributes that is useful for scheduling is the inner state of the applications in the context of multithreaded applications. The authors have suggested [20] scheduling threads based on their application's phase, whether serial or parallel. All of the above mentioned thread attributes have an effect on the performance of the scheduler.

The scheduler's task is to choose which of the ready threads to run and which resources to grant for each thread. Having many ready threads, therefore, is important for achieving high scheduling performance, since the scheduler will have more possibilities to choose from. Granting of resources to threads, however, must take into account the limited resources available in the system. For example, a scheduler might achieve better performance on a multi-core processor with a workload of many bandwidth-hungry threads, by keeping some of the cores idle, since any additional concurrently running thread will increase the number of misses in the shared cache for all of the threads and will exhaust the available memory bandwidth.

This paper presents an example of a concept: thread scheduling based on the essence of the threads and on the limited resources of the system. In particular, we present a way to maximize the number of ready threads for the scheduler when more than one multithreaded application

doi:10.4304/jcp.7.1.53-64

runs in parallel, by examining the essence of the running applications. Given the higher number of ready threads, our proposed scheduler is able to better choose which and how many threads to run, achieving better performance in various metrics.

II. Multiple Multithreaded Applications

Multithreaded applications can take advantage of the added computing ability offered by today's multiprocessors by executing in parallel on many cores. With an ever-increasing core population embedded in state-of-the-art systems [22], the use of multithreading in applications is expected to increase. In this paper, we strive to improve system performance as measured by several metrics when several multithreaded applications are run in parallel on symmetric multiprocessors, where all cores are identical, as well as on asymmetric multiprocessors [14], where some computing cores are faster than others.

When examining multithreaded applications, one can identify two types of execution phases: serial phases and parallel phases. In the serial phases, only one thread is active, whereas the parallel phases are comprised of many concurrently active threads. The number of threads as a function of time during the execution of the "equake" SPEC-OMP benchmark is shown in Fig. 1. The serial and parallel phases of the "equake" benchmark can be distinguished in Fig. 1 by the number of running threads (only one thread is active in the serial phase, and several threads are active in the parallel phase).

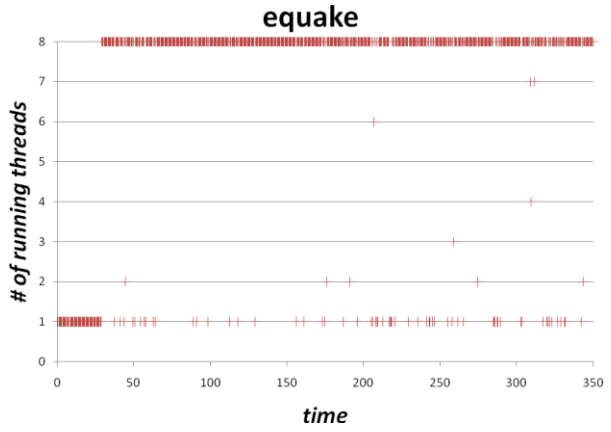


Fig. 1. The number of running threads as a function of time for the "equake" benchmark in the SPEC-OMP suite, on an 8-way multiprocessor. The serial and parallel phases can be distinguished by the number of running threads at each point of time. The sampling frequency in the figure is twice per second.

When several multithreaded applications run simultaneously on a multiprocessor, the serial thread of some applications may be available for execution together with the threads of the other applications. Fig. 2 shows an example of the four possible joint states of two multithreaded applications running simultaneously. The vertical axis represents time, and the number of active threads of each application is shown for each point of time.

Current thread assignment techniques, such as the technique used in the Linux scheduler [1], are not aware of the phases of the running applications. When multiple multithreaded applications are run in parallel, this lack of awareness may result in lower throughput, jitter in applications' runtimes (unpredictable performance), and unfairness between applications. These undesired characteristics may happen because the serial phases, which are the critical bottlenecks for the applications, compete for CPU time with the many concurrently executing parallel threads. If these serial phases were executed quickly, the application's bottlenecks would be freed, allowing the application to take advantage of the multiprocessor resources by using many threads.

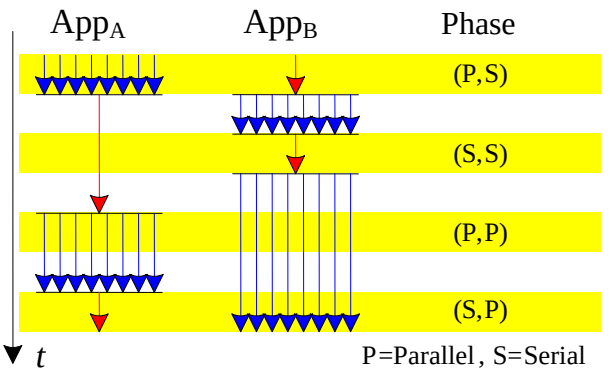


Fig. 2. Illustration of joint states of two sample applications running simultaneously.

We propose to add another dimension to the current thread assignment mechanisms by using information about the parallel and serial phases of applications. Our proposed thread assignment technique monitors the number of active threads in each application, and hence it can identify and grant higher priority to serial threads.

Fig. 3 shows transitions among the four possible joint states of two applications executing in parallel: (Serial, Serial: S,S), (Serial, Parallel: S,P), (Parallel, Serial: P,S), and (Parallel, Parallel: P,P). The large arrows on the state transition arcs denote the most likely transition. The proposed thread assignment technique, shown in Fig. 3b, favors the serial thread, thus increasing the probability for transition from (S,P) and (P,S) states to (P,P) state. Current OS schedulers (shown in Fig. 3a), however, treat the serial and parallel threads equally, thereby lengthening the time required for the serial application to transition into its parallel phase. The proposed technique improves throughput by reducing the time spent in (S,S) in which there are idle cores, resulting in a greater number of ready threads.

The new thread assignment technique corresponding to the transition likelihoods illustrated in Fig. 3b grants higher priority to applications in their serial phases in order to increase the multiprocessor throughput, improve fairness and reduce the jitter in execution runtimes. The expected improvements are quantified by a simple

analytical model. We validate our proposed techniques by experiments running on a real symmetric CMP [12] with a current version of the Linux operating system [1], and with workloads consisting of multiple multithreaded applications executing in parallel. We also validate our techniques on asymmetric structures that are emulated on the real symmetric CMP, with the addition that serial threads are granted higher priority to run on the faster cores.

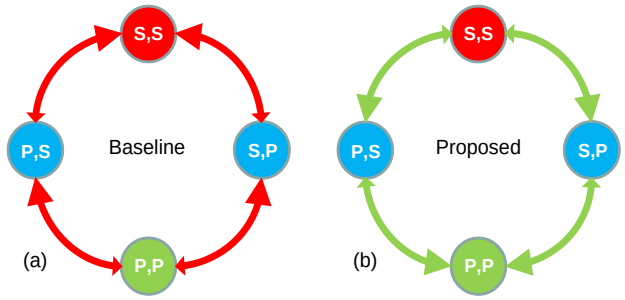


Fig. 3. Illustration of the four possible joint states of two applications running in parallel. The large arrows denote the most likely transition. Fig. 3a illustrates the states and the likely transitions in the baseline scheduler, and Fig. 3b illustrates the states and the likely transitions in the proposed scheduler, where the serial threads are favored over parallel threads.

III. Related Work

There are a number of papers addressing the scheduling of single-threaded applications on asymmetric/heterogeneous multiprocessors [14], which are based on sampling of runtime performance on the different core types. Kumar et al. [16] have proposed a scheduler for multiple single-threaded applications on a heterogeneous multiprocessor. Bower et al. [7] have shown the impact on thread scheduling in symmetric multiprocessors that become heterogeneous during runtime owing to frequency scaling, process variations and physical faults. Winter et al. [29] explored thread assignment algorithms for single-thread applications on such multiprocessors.

Other papers address the scheduling problem of a single multithreaded application running on asymmetric multiprocessors [4][6][11][15][19]. Grochowski et al. [4][11] have proposed a static scheduling mechanism, implemented at the application level, which schedules the serial phases of applications on the high performance core. Balakrishnan et al. [6] proposed a dynamic scheduler for a single multithreaded application on a heterogeneous multiprocessor. They have shown that by scheduling the serial phases on the high performance core, performance increases and the jitter in runtimes of different executions is reduced. We extend these methods [4][6][11] for multiple multithreaded programs, and address the scheduling problem that arises when there are more threads than cores in the multiprocessor.

Our results are compared with a standard Linux scheduler as a baseline environment that does not employ the proposals from previous work, since these proposals are tailored for single multithreaded applications. When more than one multithreaded application is run in parallel, a scheduler with these proposals will perform similarly to the baseline environment, and hence will only add to the complexity of evaluating our proposed techniques without offering additional insights.

Several papers explore fairness and throughput in SMT architectures [10][18][24][26]. We use and extend their throughput and fairness metrics for asymmetric multiprocessors. Other papers explore scheduling multiple multithreaded applications on symmetric systems [3][21][28]. We extend these ideas for asymmetric configurations and present the concept of prioritizing applications based on their phase of execution.

Scheduling threads based on thread attributes has been proposed in previous research [13][20][30]. Knauerhase et al. [13] suggest co-scheduling threads based on cache usage. Zhuravlev et al. [30] suggest co-scheduling threads based on memory bandwidth usage, memory controller contention and contention on the prefetching hardware. We extend these concepts and present the concept of scheduling based on thread essence, while considering the limited resources of the system on which the threads are run.

IV. Emulation Environment

All measurements in this paper were performed on an 8-core multiprocessor (HP ProLiant DL580) [12] consisting of four dual-core 2.66GHz Intel Xeon 7020 processors, 667MHz front side bus, and 8GB of DDR2 memory. SMT was disabled for better emulation of symmetric and asymmetric systems. The operating system used was Linux with kernel version 2.6.18, and is referred to in this paper as the baseline environment.

In order to emulate the asymmetric multiprocessor on the symmetric multiprocessor, the frequency (duty cycle) of the cores was changed, as was done in previous research [6][11][20]. Additionally, the Linux kernel was configured accordingly, using the Linux CPU group property CPU_POWER, to employ cores of different speeds.

The benchmarks we used in this research include the entire SPEC-OMP2001 [5] suite with the medium reference input sets, with the exception of “galgel” because of compilation difficulties in our setup.

OpenMP [23] offers various scheduling options for its parallel constructs. We altered the default OpenMP scheduling policy from static, in which each thread receives an identical portion of the workload, to dynamic, in which each thread consumes a predefined small subset of the workload and then requests additional work. This is similar to what was done in [6] and [19], and allows higher core utilization on heterogeneous multiprocessors, by reducing the time threads wait for each other.

Since the SPEC-OMP2001 benchmarks are highly parallel and represent only a small fraction of the application space, we also measure in this paper a

synthetic benchmark written by the authors. The synthetic benchmark mimics applications with an adjustable ratio of parallel to serial code. It allows us to get accurate results within a short runtime, making it practical for exploring various scheduling options for various combinations of applications running together in the system.

The synthetic benchmark consists of a loop of a mathematical calculation that fits entirely in the cache. During the course of its execution, the benchmark switches randomly between serial phases, in which there is only one active thread, and parallel phases, in which there are *n* threads, equal to the number of cores in the multiprocessor.

We model and label multithreaded programs by the ratio of parallel and serial instructions they contain, divided by the number of cores used in each phase. In the following equation, I_P and I_S denote the number of dynamic instructions executed in the parallel and serial phases respectively, *n* denotes the number of cores in the multiprocessor, and the normalization factor *k* is chosen so that one of the ratios equals one, and the other is greater than or equal to one. For simplicity, we assume identical IPC for the parallel and serial phases.

$$(ratio_{Parallel} : ratio_{Serial}) = \left(k\frac{I_P}{n} : kI_S \right) \quad (1)$$

For example, a benchmark labeled (1:1) on a symmetric CMP with no synchronization and scheduling overheads will spend roughly equal time in its parallel phases and in its serial phases.

The synthetic benchmark may be tuned so that in the long run it would mimic the parallelism behavior of applications, ranging from completely parallel applications (∞:1) to completely serial applications (1:∞). Each measurement of the synthetic benchmark lasts 60 seconds, after which the benchmark reports the total number of iterations it has completed in that time frame. The pseudo-code of the synthetic benchmark is detailed in Fig. 4.

```
while (time<60 seconds) {
    parallel_iterations = random();
    serial_iterations = parallel_iterations * ratio_Serial /(ratio_parallel * N_cores);
    in each thread { // fork
        for (i=0;i<parallel_iterations;i++)
            perform_calculation();
        calculated_iterations += parallel_iterations; //shared variable
    } //join
    for (i=0;i<serial_iterations;i++)
        perform_calculation();
    calculated_iterations += serial_iterations;
}
print "performance=",calculated_iterations/(time-start_time)
```

Fig. 4. Pseudo-code of the synthetic benchmark.

V. METHODOLOGY

This research is focused on the interactions between multiple multithreaded applications that are run in parallel. In particular, we focus on three metrics: performance, fairness, and jitter.

Measuring the performance improvement of multiple applications running in parallel in different environments (for example, environments with the same hardware but with different OS schedulers) is no trivial task [27]. It is even harder when the applications are multithreaded. Alameldeen and Wood [2] have shown that the throughput metric of IPC used in uniprocessors is not accurate for multithreaded programs in multi-core architectures. One of the reasons for this is that threads in a multithreaded program use polling when waiting for sibling threads, resulting in a different number of committed instructions in different executions of the same program. The accurate throughput metric for multithreaded programs is therefore the amount of actual work performed divided by the execution time, and not simply the IPC.

Measuring the throughput of multiple synthetic benchmarks that are running simultaneously is done by summing the number of iterations completed in each benchmark during the predefined benchmark time. The SPEC-OMP benchmarks, however, must run until completion, since they report their accurate progress only when they complete.

One way of measuring the throughput of a thread assignment mechanism for multithreaded applications is to run two applications and wait for both to finish. This method is demonstrated in Fig. 5, and is similar to the "Last" method described in [27]. While measuring with this method, we found that in many cases one application finished its execution well before the other. Since we want to measure the interactions between applications, the time segment in which only one application is active becomes irrelevant, but it does affect the results.

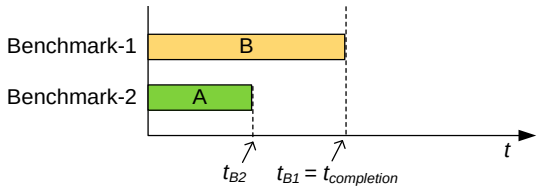


Fig. 5. Example of two multithreaded benchmarks running in parallel. Each application finishes its execution at a different time.

We handle the throughput measurement problem by running two benchmarks that perform the same work, each comprised of two applications that are run in a different order, as shown in Fig. 6. Since the work of the two benchmarks is identical, the runtimes are closer than in the previous methods. As a result, the effects of our new scheduler can be evaluated more reliably than in the other methods [27].

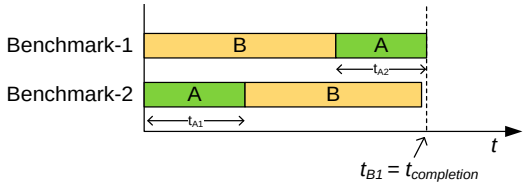


Fig. 6. Two multithreaded benchmarks, each comprising of two applications in a different order. The execution times t_{B1} and t_{B2} of the two benchmarks are similar.

The second metric evaluated in this paper is fairness. When two applications are executed in parallel, their runtimes are longer than when each application runs alone:

$$speedup_A = \frac{Performance_{A,A||B}}{Performance_{A,A}} \quad (2)$$

If both applications exhibit the same relative speedup, the system is said to be fair [10][24]. In this paper, we use the fairness metric detailed in [10], which is defined as the minimum ratio of speedups of the applications. For two applications, fairness is defined as follows:

$$Fairness_{A||B} = \min\left(\frac{speedup_A}{speedup_B}, \frac{speedup_B}{speedup_A}\right) \quad (3)$$

Fairness as defined above can be in the range of zero to one, corresponding to completely unfair and to completely fair, respectively. We calculate the speedup of application "A" in equation (2) as the time required to execute the application alone on the multiprocessor, divided by the average duration of application "A" in the configuration shown in Fig. 6:

$$Speedup_A = \frac{t_{A,alone}}{\frac{1}{2}\left(t_{A_1} + t_{A_2}\right)} \quad (4)$$

The third metric we consider is jitter in execution runtimes. Balakrishnan et al. [6] have already shown that operating system schedulers in asymmetric multiprocessors present unpredictable application runtimes for a single multithreaded application. In this paper, we quantify runtime jitter by measuring the standard deviation of the normalized execution times of the workload in *N* experiments of the same benchmark:

$$Jitter_A = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\frac{t_{A,n}}{t_{A,avg}} - 1\right)^2} \quad (5)$$

VI. ANALYSIS

In this section, we analyze the performance, fairness, and jitter metrics for multiple multithreaded applications running in parallel. These applications have one active thread in their serial phases and *n* active threads in their parallel phases, which is also equal to the number of cores in the multiprocessor. This assumption does not always hold in reality, since not all threads reach their barriers at the same time. When using dynamic work distribution with workloads consisting of long parallel phases, however, the effects of having less than *n* active threads in the parallel phases can be neglected. The applications in this model may differ in their parallel/serial ratios. The performance figures in this section are normalized to the performance of one thread on one small core.

We consider an asymmetric multiprocessor with *n* cores: one of the cores is larger as in [19] and faster by a factor (*a*). For simplicity, we assume that the performance factor (*a*) is identical for all workloads. The results in this section can be derived for symmetric multiprocessors by assigning *a*=1.

When only one thread is running in the system, we assume that the scheduler will schedule the thread on the higher performance core. The performance of a serial thread on an idle asymmetric multiprocessor is thus given by:

$$Perf_{serial} = a \quad (6)$$

When a multithreaded application with *n* threads runs on a multiprocessor with *n* cores, we assume that the scheduler will distribute the threads on the cores so that no core will be left idle. This assumption makes sense for sufficiently small values of (*a*), or ($1 \le a < 2$). For higher values of (*a*) the scheduler could potentially achieve better performance by scheduling more threads on the larger core and leave some of the smaller cores idle. For simplicity, we assume that (*a*) is sufficiently small ($1 \le a < 2$) in our analysis. The performance of a parallel application on an idle asymmetric multiprocessor is thus given by:

$$Perf_{parallel} = n - 1 + a \quad (7)$$

We consider the case where two concurrently running applications, App_A and App_B, have *n* active threads in their parallel phases.

When both applications are serial, the scheduler has two options for scheduling. In the first option, one application is scheduled to run on the larger core, and the other application is scheduled to run on a small core. In the second option both applications can run on the larger core. Since we assume that (*a*) is sufficiently small ($1 \le a < 2$) this option is not considered in our analysis. The maximum speedup in this case will be achieved when the serial application runs on the larger core:

$$Speedup_{(S,S),\max} = 1 \quad (8)$$

The minimum speedup will be achieved when the serial application runs on a small core:

$$Speedup_{(S,S),\min} = \frac{1}{a} \quad (9)$$

Given that each application has an equal chance to run on the larger core, the speedup is as follows:

$$Speedup_{(S,S),avg} = \frac{1}{2}\cdot\frac{1}{a} + \frac{1}{2}\cdot 1 = \frac{1}{2}\left(1+\frac{1}{a}\right) \quad (10)$$

When both applications are in their parallel phases, there are *2n* running threads that compete for *n* cores. The Linux scheduler will schedule two threads on each core, thereby slowing down each application by a factor of two, assuming that the threads have equal priority and are not IO bound.

The maximum speedup for a parallel application will therefore be achieved when two threads of the parallel application are scheduled on the larger core, and each of the other threads of the parallel application run in conjunction with another thread on a small core:

$$Speedup_{(P,P),\max} = \frac{(n-2)\cdot\frac{1}{2} + 2\cdot\frac{a}{2}}{n-1+a} = \frac{n-2+2a}{2(n-1+a)} \quad (11)$$

The minimum speedup for a parallel application will be achieved when each of the threads of the parallel application runs in conjunction with another thread on a small core, and none of the threads run on the faster core:

$$Speedup_{(P,P),\min} = \frac{(n)\cdot\frac{1}{2} + 0\cdot\frac{a}{2}}{n-1+a} = \frac{n}{2(n-1+a)} \quad (12)$$

In the average case, each of the parallel applications will exhibit a speedup of 0.5:

$$Speedup_{(P,P),avg} = \left(\frac{n}{2n}\frac{n-1}{2n-1}\right)\frac{(n-2)\cdot\frac{1}{2} + 2\cdot\frac{a}{2}}{n-1+a} + \left(2\frac{n}{2n}\frac{n}{2n-1}\right)\frac{(n-1)\cdot\frac{1}{2} + 1\cdot\frac{a}{2}}{n-1+a} + \left(\frac{n}{2n}\frac{n-1}{2n-1}\right)\frac{(n-2)\cdot\frac{1}{2} + 2\cdot\frac{a}{2}}{n-1+a} = \frac{1}{2} \quad (13)$$

When one of the applications is serial and the other is parallel, there are *n+1* threads that are to be scheduled on *n* cores. Out of the *n+1* threads, two threads will share a single core, and *n-1* threads will each have their own core. In the worst case for the serial application, it will be assigned to run with another thread on a small core:

$$Speedup_{(S,P),\min} = \frac{1}{2}\cdot\frac{1}{a} = \frac{1}{2a} \quad (14)$$

In the best case for the serial application, it will exhibit no slowdown as it will be scheduled to run on the large core by itself:

$$Speedup_{(S,P),\max} = 1 \quad (15)$$

For simplicity, we assume that all threads have equal probability to execute on the two-thread core and on the one-thread core. The probability of the serial thread sharing its core with another thread is therefore $2(n-1)^{-1}$, and the probability that this core is the larger core is n^{-1}. The average performance of a serial application when running concurrently with a parallel application is thus:

$$Speedup_{(S,P),avg} = \frac{2}{n+1}\left(\frac{1}{n}\cdot\frac{1}{2} + \frac{n-1}{n}\cdot\frac{1}{2a}\right) + \frac{n-1}{n+1}\left(\frac{1}{n}\cdot 1 + \frac{n-1}{n}\cdot\frac{1}{a}\right) = \frac{n-1+a}{a(n+1)} \quad (16)$$

When a parallel application is running on an asymmetric multiprocessor concurrently with a serial application, the maximum speedup is achieved when the serial thread is scheduled together with one of the parallel threads on one of the small cores:

$$Speedup_{(P,S),\max} = \frac{1\cdot\frac{1}{2} + 1\cdot a + (n-2)\cdot 1}{n-1+a} = \frac{n-\frac{3}{2}+a}{n-1+a} \quad (17)$$

The minimum speedup is achieved when the serial thread is scheduled alone on the larger core:

$$Speedup_{(P,S),\min} = \frac{(n-2)\cdot 1 + 2\cdot\frac{1}{2} + 0\cdot a}{n-1+a} = \frac{n-1}{n-1+a} \quad (18)$$

The average performance of the parallel application when running simultaneously with a serial application is given by:

$$Perf_{(P,S),avg} = \frac{1}{n-1+a} \cdot \left(\begin{array}{l} \frac{2}{n+1}\left(\frac{1}{n}\cdot(n-1+\frac{a}{2}) + \frac{n-1}{n}\cdot(n-\frac{3}{2}+a)\right) + \\ \frac{n-1}{n+1}\left(\frac{1}{n}\cdot(n-1) + \frac{n-1}{n}\cdot(n-2+a)\right) \end{array} \right) = \frac{n}{n+1} \quad (19)$$

The minimum, maximum and average speedups as calculated in the above equations are summarized in Table 1 for asymmetric multiprocessors.

TABLE 1. SPEEDUPS (MIN, AVERAGE, MAX) FOR APPLICATION "A" IN THE BASELINE ENVIRONMENT ON THE ASYMMETRIC MULTIPROCESSOR. *n*=NUMBER OF CORES. *a*=PERFORMANCE RATIO OF THE LARGE CORE.

Case (A,B)	Minimum Speedup	Average Speedup	Maximum Speedup	Maximum/ Minimum
(S,S)	$\frac{1}{a}$	$\frac{1}{2}\left(1+\frac{1}{a}\right)$	1	a
(S,P)	$\frac{1}{2a}$	$\frac{n-1+a}{a(n+1)}$	1	$2a$
(P,S)	$\frac{n-1}{n-1+a}$	$\frac{n}{(n+1)}$	$\frac{n-3/2+a}{n-1+a}$	$\frac{n-3/2+a}{n-1}$
(P,P)	$\frac{n}{2(n-1+a)}$	$\frac{1}{2}$	$\frac{n-2+2a}{2(n-1+a)}$	$\frac{n-2+2a}{n}$

Table 2 shows the speedups calculated for symmetric multiprocessors (a=1).

TABLE 2. SPEEDUPS (MIN, AVERAGE, MAX) FOR APPLICATION "A" IN THE BASELINE ENVIRONMENT ON THE SYMMETRIC MULTIPROCESSOR. *n*=NUMBER OF CORES.

Case (A,B)	Minimum Speedup	Average Speedup	Maximum Speedup	Maximum/ Minimum
(S,S)	1	1	1	1
(S,P)	$\frac{1}{2}$	$\frac{n}{n+1}$	1	2
(P,S)	$\frac{n-1}{n}$	$\frac{n}{(n+1)}$	$\frac{n-1/2}{n}$	$\frac{n-1/2}{n-1}$
(P,P)	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1

Fairness is calculated according to (3). When two applications are both in their serial phase, the lower bound for fairness is given by dividing the minimum and maximum speedups of the state (S,S):

$$Fairness_{(S,S)} = \frac{Speedup_{(S,S),\min}}{Speedup_{(S,S),\max}} = \frac{1}{a} \quad (20)$$

Similarly, the lower bound for fairness when both applications are in their parallel phases is given by dividing the minimum and maximum speedups of the state (P,P):

$$Fairness_{(P,P)} = \frac{n}{n-2+2a} \quad (21)$$

When one application is serial and the other is parallel, there are two cases for fairness. In the first case, the lower bound for fairness is given by dividing the minimum speedup in the state (S,P) by the maximum speedup in state (P,S):

$$Fairness_{(S,P),1} = \min\left(\frac{Speedup_{(S,P),\min}}{Speedup_{(P,S),\max}}, \frac{Speedup_{(P,S),\max}}{Speedup_{(S,P),\min}}\right) = \frac{n-1+a}{2a\left(n-3/2+a\right)} \quad (22)$$

In the second case, the lower bound for fairness is given by dividing the minimum speedup in state (P,S) by the maximum speedup in state (S,P):

$$Fairness_{(S,P),2} = \min\left(\frac{Speedup_{(S,P),\max}}{Speedup_{(P,S),\min}}, \frac{Speedup_{(P,S),\min}}{Speedup_{(S,P),\max}}\right) = \frac{n-1}{n-1+a} \quad (23)$$

The lower bounds of the fairness metric as calculated in the above equations for the baseline environment are summarized in Table 3.

TABLE 3. LOWER BOUND FOR FAIRNESS IN THE BASELINE ENVIRONMENT.

(S,S)	(S,P) case 1	(S,P) case 2	(P,P)
$\frac{1}{a}$	$\frac{n-1+a}{2a\left(n-3/2+a\right)}$	$\frac{n-1}{n-1+a}$	$\frac{n}{n-2+2a}$

The minimum and maximum speedups in the different joint phases, as detailed in Table 1, and the equations for the lower bound of fairness, as detailed in Table 3, indicate that as the ratio between the performance of the cores in the asymmetric multiprocessor (a) increases, the lower bound for fairness decreases and the jitter between the runtimes increases.

We extend the analysis in this section for k>2 multithreaded applications running in parallel. The extension results are detailed in Table 4. The "Serial" or "Parallel" rows in the table show the speedups for a serial or parallel phase of an application under all possible phases of the other applications running in parallel, in comparison with it running alone. The analysis predicts that as the number of applications (k) that are run in parallel increases, the possible jitter widens.

The probability of having idle cores decreases exponentially as more parallel applications are run in parallel (S,S,S,…). Consequently, the throughput gains of using our mechanism are expected to decrease as the number of parallel applications that are run in parallel increases.

TABLE 4. MINIMUM AND MAXIMUM SPEEDUPS FOR 2<=k<=n APPLICATIONS RUNNING IN PARALLEL.

Phase	Minimum Speedup	Maximum Speedup	Maximum / Minimum
Serial	$\frac{1}{ka}$	1	ka
Parallel	$\frac{n}{k(n-1+a)}$	$\frac{n-1+a-\frac{k-1}{2}}{n-1+a}$	$k\frac{n-1+a-\frac{k-1}{2}}{n}$

In this section we presented analytic tools to predict the speedup and fairness of applications running in parallel. The analysis predicts that applications may exhibit significant slowdowns due to the contentions with other running applications. One of the factors shown to affect the slowdowns is the joint-phases of the applications that are currently running. These joint-phases are hard to predict in advance, potentially resulting in unpredictable performance of the applications. Moreover, each application may exhibit different slowdowns, as shown in the lower bounds of the fairness equations. The analysis also predicts that as the asymmetry widens, the possible jitter increases and the lower bound of fairness decreases.

VII. PROPOSED ALGORITHM

We propose a new thread assignment algorithm that aims to improve performance, improve fairness and reduce the jitter in execution runtimes. The proposed algorithm examines the essence of the running applications and grants higher scheduling priority to serial threads. As a result, when a serial thread is executed concurrently with a parallel application, the serial thread is granted a core for itself, and the threads of the parallel application will compete for the remaining cores. The scheduling mechanism results in the speedups shown in Table 5, which differs from Table 1 in states (S,P) and (P,S).

TABLE 5. MINIMUM AND MAXIMUM SPEEDUPS OF APPLICATION "A" FOR THE PROPOSED THREAD ASSIGNMENT TECHNIQUE ON ASYMMETRIC MULTIPROCESSORS.

Case (A,B)	Minimum Speedup	Average Speedup	Maximum Speedup	Maximum/ Minimum
(S,S)	$\frac{1}{a}$	$\frac{1}{2}\left(1+\frac{1}{a}\right)$	1	a
(S,P)	1	1	1	1
(P,S)	$\frac{n-1}{n-1+a}$	$\frac{n-1}{n-1+a}$	$\frac{n-1}{n-1+a}$	1
(P,P)	$\frac{n}{2(n-1+a)}$	$\frac{1}{2}$	$\frac{n-2+2a}{2(n-1+a)}$	$\frac{n-2+2a}{n}$

For the symmetric case (a=1), our analysis predicts identical minimum and maximum execution times for all states, so that jitter will be minimized and fairness between applications will improve using our proposed scheduler.

It is expected that the number of ready threads will be greater on average with the proposed scheduler, since the serial threads will execute faster and allow the parallel phases to start executing sooner. The proposed scheduler will thus be able to increase the utilization of the multiprocessor by scheduling additional threads that will run concurrently.

In state (S,S) on the asymmetric multiprocessor, there are two active serial threads but only one of them is granted the larger core. This presents jitter in execution times, which could be avoided, for example, by the method proposed by Fedorova et al. [8] at the expense of many thread migrations, or by the method presented by Li et al. [17]. Another possible method is to grant priority for processing power per application and not per thread. State (P,P) is similar, and the jitter in this state could also be avoided by using similar methods.

The predicted speedups of the proposed scheduler for more than two applications running in parallel are detailed in Table 6.

The Linux scheduler was extended to support the proposed algorithm. The proposed scheduler continuously monitors the number of ready threads in each thread group, and hence can detect whether an application is in its parallel phase or in its serial phase. This is performed in O(1) time whenever a thread changes its ready state. In our implementation, a thread group is considered parallel when it has more than two ready threads, and is considered serial otherwise. We chose two as the threshold since we noticed that an

OpenMP application frequently switched between one and two active threads.

TABLE 6. SPEEDUPS FOR 2<=*k*<=*n* APPLICATIONS RUNNING IN PARALLEL USING THE PROPOSED SCHEDULER.

Phase	Minimum Speedup	Maximum Speedup	Maximum / Minimum
Serial	$\frac{1}{a}$	1	a
Parallel	$\frac{n}{k(n-1+a)}$	$\frac{n-k+1}{n-1+a}$	$k\frac{n-k+1}{n}$

The scheduler was also extended to grant higher priority to serial threads. In Linux, each thread has a property known as dynamic priority. When the dynamic priority figure of a thread is lower, the thread is granted more CPU time. The priority of the thread was therefore boosted by subtracting ten [1] from its dynamic priority property.

The load balancer of the Linux kernel was extended as well; the baseline scheduler will not migrate a running thread, and will not migrate a ready thread from a slow core to a fast core if it is the only running thread on the slow core. These were changed to allow for better load balancing on asymmetric multiprocessors.

When at least two applications are in their parallel phases, and each has a number of active threads that is at least equal to the number of cores in the system, the applications compete with each other without any throughput gains. This competition, which is favored by our proposed technique for maximizing the number of ready threads, results in many unnecessary context switches that thrash the cache and lower the overall throughput of the system.

Our proposed technique strives to avoid this situation by considering the limited resources of the system, and boosts the priority of the application that was the first to enter its parallel phase. We call this mechanism "seniority boost", as the scheduler chooses the senior application and boosts its priority. This mechanism is similar to gang scheduling [9][25]. When using this mechanism, the application with the seniority boost is expected to finish its parallel phase sooner, while the system exhibits fewer context switches. When one of the applications finishes its parallel phase, the system transitions to one of the joint states (P,S) or (S,P) and the seniority boost is removed. In order to avoid starvation, following a timeout in state (P,P) the seniority boost is removed and applied to the other application.

The baseline Linux scheduler's thread migration policy has also been revised. Threads whose applications become serial are automatically rescheduled on the idlest core and are granted more priority. In asymmetric configurations, the high priority given to these threads will usually result in migration to the higher performance core.

The asymmetric multiprocessor is emulated by changing the frequency (duty cycle) of seven out of eight cores in our symmetric multiprocessor, as was done in [6] and [11]. In our case, we chose a=2, so the frequency of seven of the eight cores was halved. Additionally, we configured the scheduler to treat the larger core as having more performance by using the Linux CPU group property "CPU_POWER". As a result, the scheduler attempted to schedule more work on the larger core.

The proposed scheduler will require additional changes to perform well when there are more applications than cores, since serial threads may dominate the computing resources. Such changes may include a timeout for the bonus granted to serial threads.

VIII. EXPERIMENTAL RESULTS

The idle time percentage measured for two synthetic benchmarks running in parallel decreased as expected, from 20% to 17.2% (reduction by 14%) in the symmetric configuration, and from 25.6% to 22.8% (reduction by 10.9%) in the asymmetric configuration. This is in line with our expectations that the multiprocessor's utilization will be increased with the proposed scheduler. Throughput improved by 3% and 4.5% respectively for the symmetric and asymmetric configurations, as shown in Table 7 for the asymmetric configuration.

TABLE 7. MEASURED SPEEDUP OF TWO CONCURRENTLY RUNNING SYNTHETIC BENCHMARKS USING THE PROPOSED TECHNIQUE ON AN ASYMMETRIC MULTIPROCESSOR CONFIGURATION (a=2).

	(∞:1)	(8:1)	(4:1)	(2:1)	(1:1)	(1:2)	(1:4)	(1:8)	(1:∞)
(∞:1)	1%								
(8:1)	-1%	1%							
(4:1)	-1%	1%	1%						
(2:1)	0%	-1%	4%	4%					
(1:1)	-2%	1%	3%	4%	7%				
(1:2)	0%	1%	3%	5%	8%	7%			
(1:4)	-2%	2%	0%	6%	9%	11%	8%		
(1:8)	-2%	1%	3%	8%	7%	15%	18%	3%	
(1:∞)	-2%	2%	3%	6%	12%	16%	17%	10%	12%
AVG	-1%	1%	2%	4%	5%	7%	8%	7%	8%
Average speedup of all dual benchmarks: +4.5%									

Fig. 7 shows a contour graph of the speedup in the symmetric multiprocessor. Each axis represents an application, ranging from completely serial (1:∞) to completely parallel (∞:1). The data in the graph corresponds to the speedup of the two applications running in parallel on a symmetric multiprocessor with the proposed scheduler, in comparison to the baseline scheduler. Peak speedup is achieved by the combination of benchmark (1:1) with a similar benchmark (1:1). Speedups decrease monotonically when moving away from this peak. The expected speedups of the highly parallel SPEC-OMP2001 benchmarks should roughly correspond to the (∞:1) and (8:1) benchmarks, which are between 0%–4% in the symmetric configuration.

The throughput gains for three synthetic benchmarks running in parallel improved with the new scheduler by 1% in the symmetric configuration and by 1.87% in the asymmetric configuration. These results were expected, as the measured CPU idle time in the symmetric case for

three benchmarks was 13.62% (or 12.25% with the proposed scheduler) in comparison to 20% for two applications (or 17.2% with the proposed scheduler). With additional applications running in parallel, the average idle time decreases, leaving less room for improvement for our proposed scheduler.

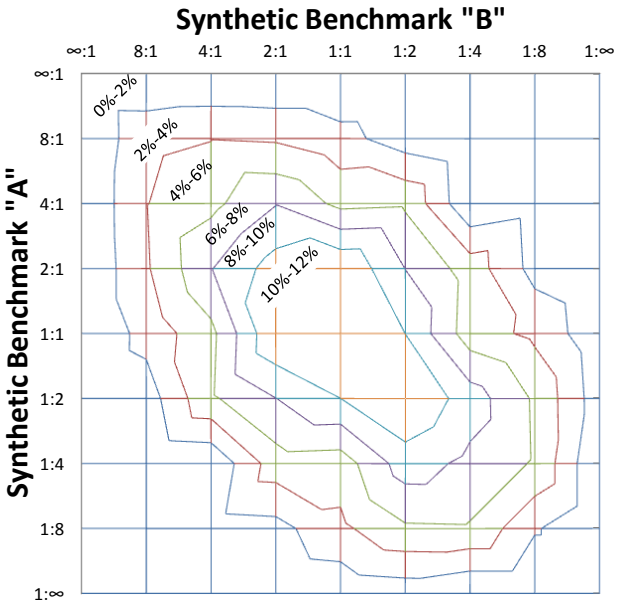


Fig. 7. Experimental contours of the speedup of two concurrently running synthetic benchmarks when the proposed technique is used on a symmetric multiprocessor configuration (a=1).

Table 8 shows the speedups for the SPEC-OMP2001 benchmarks with the proposed thread assignment technique. The measurements were performed according to the method shown in Fig. 6. The speedup exhibited by the highly parallel SPEC-OMP benchmarks averaged 1.5% in the symmetric multiprocessor, and 3.5% in the asymmetric multiprocessor. The "apsi" benchmark showed significant improvement because it had many phase shifts between parallel/serial phases. Since our proposed mechanism reacts fast to these frequent phase shifts, the bottlenecks of "apsi" were freed faster, and hence "apsi" achieved greater speedups.

TABLE 8. MEASURED SPEEDUP OF TWO CONCURRENTLY RUNNING SPEC-OMP2001 BENCHMARKS USING THE PROPOSED TECHNIQUE ON AN ASYMMETRIC MULTIPROCESSOR CONFIGURATION (a=2).

	wup									
wupwise	2%	swi								
swim	8%	1%	mgr							
mgrid	4%	4%	4%	app						
applu	2%	-2%	3%	1%	equ					
equake	3%	0%	4%	0%	0%	aps				
apsi	12%	15%	7%	12%	9%	16%	gaf			
gafort	1%	3%	2%	-2%	2%	7%	0%	fma		
fma3d	1%	5%	3%	-3%	0%	9%	3%	3%	art	
art	2%	0%	4%	3%	-1%	15%	-1%	2%	-1%	amm
ammp	3%	1%	4%	-2%	4%	13%	0%	4%	1%	1%
average	4%	3%	4%	1%	2%	11%	1%	3%	2%	3%
Average speedup of all dual benchmarks: 3.5%										

When running two SPEC-OMP benchmarks concurrently as in Fig. 6, compared with running the benchmarks sequentially only one application at a time, throughput increases by 36% on average in the baseline scheduler, and by 38% on average on the proposed scheduler. In some cases, however, the "ammp" and "wupwise" benchmarks ran faster sequentially in the baseline and proposed schedulers compared to running concurrently. This slowdown was caused by the contention between the threads on the system resources. A scheduler that is aware of the system resource limits could potentially mitigate this slowdown by running only the threads that the system has enough resources for them to run.

The throughput gains for three synthetic benchmarks running in parallel improved with the new scheduler by 1% in the symmetric configuration and by 1.87% in the asymmetric configuration. These results were expected, as the measured CPU idle time in the symmetric case for three benchmarks was 13.62% (or 12.25% with the proposed scheduler) in comparison to 20% for two applications (or 17.2% with the proposed scheduler). With additional applications running in parallel, the average idle time decreases, leaving less room for improvement for our proposed scheduler.

The jitter for the synthetic benchmarks multiplied by 1000 is shown in Table 9, and was reduced on average by 60% in the symmetric case and by 35% in the asymmetric case. The jitter, measured on five runs of "equake" and "art" as an example, was almost eliminated in the symmetric case and was halved in the asymmetric case. Fairness has improved as well in almost all benchmarks. Notably, the lower bound of fairness as measured in five runs of "equake" and "art" improved by 26%.

TABLE 9. THE AVERAGE FAIRNESS AND JITTER METRICS WITH THE BASELINE AND PROPOSED ENVIRONMENTS FOR THE SYNTHETIC BENCHMARKS AND FOR SPEC-OMP (5 EXECUTIONS OF "ART" & "EQUAKE").

Benchmark	Scheduler	Symmetric (A=1)		Asymmetric (A=2)	
		Fairness	Jitter	Fairness	Jitter
Synthetic	Baseline	75.9%	9.07	87.5%	38.74
	Proposed	90.7%	3.66	88.7%	25.12
	Improvement	**19.5%**	**59.7%**	**1.4%**	**35.1%**
SPEC-OMP	Baseline	79.6%	1.13	49.3%	1.90
	Proposed	78.5%	0.13	62.1%	0.94
	Improvement	**-1.4%**	**88.1%**	**25.9%**	**50.5%**

IX. CONCLUSIONS AND FUTURE WORK

In multiprocessors running multiple multithreaded applications, thread scheduling may be performed based on the essence of the threads and on the resource limitations of the system. As an example, our proposed thread assignment mechanism examines the essence of the threads and favors serial phases of applications over parallel phases, in a first attempt to optimize multiple multithreaded applications running in parallel on symmetric and asymmetric multiprocessors.

Detailed analysis for several multithreaded applications running simultaneously shows potential for improvements in throughput, fairness and the jitter metrics. In particular, when two multithreaded scientific applications (SPEC-OMP2001) are run on symmetric as

well as on asymmetric multiprocessors, analytical and experimental results show improvements in all metrics; the jitter in execution runtimes decreased by as much as 88%, throughput in some cases increased by more than 16%, and the fairness metric improved by up to 26%.

The experiments in this paper were performed on a real system, using official benchmarks and a modern operating system (Linux kernel 2.6.18) with our extensions. The concepts of this work could easily be implemented in today's state-of-the-art multiprocessor operating systems, as implemented in our experimental system, and could show immediate performance gains. Moreover, the concepts could be used in today's grid architectures to better exploit the computing power of shared memory nodes by scheduling several multithreaded workloads at once.

There are various architectural implications for this work. First, chip architects designing asymmetric multiprocessors can use the analysis presented in this paper for predicting the effects of asymmetry on various system metrics. We found that as asymmetry between the cores widens, fairness worsens and jitter between execution runtimes increases. Second, exploiting asymmetry requires faster thread migration techniques. A first step towards faster migration techniques was presented in this research, and future work should continue to explore how to react fast to the constant and continuous changes in the available system resources. We believe that in future designs, hardware may assist the OS in performing these migrations, as opposed to current designs in which the OS migrates threads without any hardware assistance. Third, the performance improvements presented in this paper for asymmetric structures show that asymmetry presents even greater performance potential over symmetric designs than predicted by previous research.

Future work can extend the analysis to take into account the distribution of phase changing during the runtime of applications. Additionally, the way multithreaded programs were modeled in this paper, with either one active thread or *n* active threads, could be extended to include the whole range from one to *n*. Such extensions could consequently be used to improve system metrics even further, even on current symmetric architectures. The analysis could also be extended to support various configurations of asymmetric multiprocessors, such as more than two types of cores. Furthermore, the analysis could take into account different speedups for different applications on each core type.

This work also provides insights into a multitude of future research issues in the area of multithreaded application handling in CMP. Essence of threads can be further explored to point out the most influential attributes that affect scheduling. Such attributes may include cache usage, memory bandwidth, dependencies on other threads, IO access patterns and more.

Scheduling according to resource limitations should be explored further. We have shown that in some cases it is better to run multithreaded applications sequentially than to run them concurrently. Future schedulers should avoid scheduling threads if they will use more resources than are currently available in the system. This will allow better usage of the system resources while conserving power.

ACKNOWLEDGMENT

The authors would like to thank Dror Feitelson, Idit Keidar, Avi Mendelson and Ronny Ronen for their insightful comments. They would also like to thank Andrey Gelman and Niv Aibester for their help in setting up the emulation environment.

REFERENCES

[1] J. Aas, "Understanding the Linux 2.6.8.1 CPU Scheduler," SGI, 2005.

[2] A.R. Alameldeen and D.A. Wood, "IPC Considered Harmful for Multiprocessor Workloads," in IEEE Micro Jul–Aug 2006.

[3] T.E. Anderson, B.N. Bershad, E.D. Lazowska, and H.M. Levy, "Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism," in ACM Transactions on Computer Systems, Vol. 10, No. 1, February 1992.

[4] M. Annavaram, E. Grochowski, and J. Shen, "Mitigating Amdahl's Law Through EPI Throttling," in Proc. of the 35th ISCA, June 2005.

[5] V. Aslot, M. Domeika, R. Eigenmann, G. Gaertner, W.B. Jones, and B. Parady, "SPEComp: A New Benchmark Suite for Measuring Parallel Computer Performance," in Proc. of WOMPAT, 2001.

[6] S. Balakrishnan, R. Rajwar, M. Upton, and K. Lai, "The Impact of Performance Asymmetry in Emerging Multicore Architectures," in Proc. of the 35th ISCA, June 2005.

[7] F.A. Bower, D.J. Sorin, and L.P. Cox, "The Impact of Dynamically Heterogeneous Multicore Processors on Thread Scheduling," in IEEE Micro May–June 2008.

[8] A. Fedorova, D. Vengerov, and D. Doucette, "Operating System Scheduling On Heterogeneous Core Systems," in Proc. of OSHMA workshop, 16th PACT, 2007.

[9] D.G. Feitelson and L. Rudolph, "Evaluation of Design Choices for Gang Scheduling using Distributed Hierarchical Control," in J. Parallel & Distributed Computing 35(1), May 1996.

[10] R. Gabor, S. Weiss, and A. Mendelson, "Fairness and Throughput in Switch on Event Multithreading," in Proc. of the 39th International Symposium on Microarchitecture, 2006.

[11] E. Grochowski, R. Ronen, J. Shen, and H. Wang, "Best of Both Latency and Throughput," in Proc. of the 22nd ICCD, October 2004.

[12] HP Technical Document DA-12195, "HP Proliant DL580 Generation 3 (G3)," Version 30 – May 2007: http://h18004.www1.hp.com/products/quickspecs/12195_div/12195_div.pdf.

[13] R. Knauerhase, P. Brett, B. Hohlt, T. Li, and S. Hahn, "Using OS Observations to Improve Performance in Multicore Systems," in IEEE Micro 28, 2008.

[14] M. D. Hill and M. R. Marty, "Amdahl's Law in the Multicore Era," in IEEE Computer, July 2008.

[15] D. Koufaty, D. Reddy, and S. Hahn, "Bias Scheduling in Heterogeneous Multi-core Architectures," in Proc. of EuroSys 2010.

[16] R. Kumar, D. Tullsen, P. Ranganathan, N. Jouppi, and K. Farkas, "Single-ISA Heterogeneous Multi-core Architectures for Multithreaded Workload Performance," in Proc. of the 31st ISCA, June 2004.

[17] T. Li, P. Brett, R. Knauerhase, D. Koufaty, D. Reddy, and S. Hahn, "Operating System Support for Overlapping-ISA Heterogeneous Multi-core Architectures," in Proc. of the 16th HPCA, January 2010.

[18] K. Luo, J. Gummaraju, and M. Franklin, "Balancing Throughput and Fairness in SMT Processors," in Proc. of the ISPASS, pages 164–171, 2001.

[19] T.Y. Morad, U.C. Weiser, A. Kolodny, M. Valero, and E. Ayguadé, "Performance, Power Efficiency, and Scalability of Asymmetric Cluster Chip Multiprocessors," in Computer Architecture Letters, Vol. 4, 2005.

[20] T.Y. Morad, A. Kolodny, and U.C. Weiser, "Scheduling Multiple Multithreaded Applications on Asymmetric and Symmetric Chip Multiprocessors," in the 3rd International Symposium on Parallel Architectures, Algorithms and Programming (PAAP'10), Dalian, 2010.

[21] D.S. Nikolopoulos, C.D. Antonopoulos, I.E. Venetis, P.E. Hadjidoukas, E.D. Polychronopoulos, and T.S. Papatheodorou, "Achieving Multiprogramming Scalability on Intel SMP Platforms: Nanothreading in the Linux Kernel," in PARCO 1999.

[22] K. Olukotun and L. Hammond, "The Future of Microprocessors," in ACM Queue, Vol. 3, No. 7, 2005.

[23] OpenMP Architecture Review Board, "OpenMP Application Program Interface," http://www.openmp.org, version 2.5, May 2005.

[24] S.E. Raasch and S.K. Reinhardt, "Applications of Thread Prioritization in SMT Processors," in Proc. 1999 Workshop on Multithreaded Execution and Compilation, 1999.

[25] U. Schwiegeishohn, R. Yahyapour, "Improving First-Come-First-Serve Job Scheduling by Gang Scheduling," in IPPS'98 Workshop, March 1998.

[26] A. Snavely, D. Tullsen, and G. Voelker, "Symbiotic job scheduling with priorities for a simultaneous multithreading processor," in Proc. of the 2002 ACM SIGMETRICS, 2002.

[27] J. Vera, F.J. Cazorla, A. Pajuelo, O.J. Santana, E. Fernández and M. Valero, "FAME: FAirly MEasuring Multithreaded Architectures," in IEEE-ACM PACT Conference, Brasov, 2007.

[28] I.E. Venetis, D.S. Nikolopoulos, and T.S. Papatheodorou, "A Transparent Operating System Infrastructure for Embedding Adaptability to Thread-Based Programming Models," in EuroPar, 2001.

[29] J.A. Winter and D.H. Albonesi, "Scheduling Algorithms for Unpredictably Heterogeneous CMP Architectures," in Proc. of the 38th DSN, June 2008.

[30] S. Zhuravlev, S. Blagodurov, and A. Fedorova, "Addressing Shared Resource Contention in Multicore Processors via Scheduling," in Proc. of ASPLOS'10, New York, 2010.

Tomer Y. Morad is a Ph.D. Candidate at the Technion – Israel Institute of Technology. He is also the Chief Technology Officer of transSpot Ltd. Tomer earned his B.Sc. and M.Sc. in Electrical Engineering from the Technion in 2001 and 2005 respectively. His main research interests are asymmetric cluster chip multiprocessors and operating system schedulers.

Avinoam Kolodny is an associate professor of electrical engineering at the Technion – Israel Institute of Technology. He joined Intel after completing his doctorate in microelectronics at the Technion in 1980. During twenty years with the company he was engaged in diverse areas including non-volatile memory device physics, electronic design automation and organizational development. He pioneered static timing analysis of processors as the lead developer of the CLCD tool, served as Intel's corporate CAD system architect in California during the co-development of the RLS system and the 486 processor, and was manager of Intel's performance verification CAD group in Israel. He has been a member of the Faculty of Electrical Engineering at the Technion since 2000. His current research is focused primarily on interconnect issues in VLSI systems, covering all levels from physical design of wires to networks on chip and multi-core systems.

Uri C. Weiser is a visiting Professor at the Electrical Engineering department, the Technion IIT and is in the advisory board of numerous startups. He received the bachelor and master degrees in EE from the Technion and Ph.D. in CS from the University of Utah, Salt Lake City. Professor Weiser worked at Intel from 1988-2006. At Intel, Weiser initiated the definition of the Pentium® processor, drove the definition of Intel's MMX™ technology, invented the Trace Cache, co-managed the a new Intel Microprocessor Design Center at Austin, Texas and formed an Advanced Media applications research activity. Weiser was appointed an Intel Fellow in 1996, in 2002 he became an IEEE Fellow and in 2005 an ACM Fellow. Prior to his career at Intel, Professor Weiser worked for the Israeli Department of Defense as a research and system engineer and later with National Semiconductor Design Center in Israel, where he led the design of the NS32532 microprocessor. Professor Weiser was an Associate Editor of IEEEMicro Magazine (1992-2004) and is an Associate Editor of Computer Architecture Letters.

A Game Theoretic Resource Allocation Model Based on Extended Second Price Sealed Auction in Grid Computing

Weifeng Sun
School of Software, Dalian University of Technology, Dalian, China
Email: wfsun@dlut.edu.cn

Qiufen Xia, Zichuan Xu, Mingchu Li, Zhenquan Qin
School of Software, Dalian University of Technology, Dalian, China
Email: {xiaqiufen, zichuanxu.mail}@gmail.com, mingchul@dlut.edu.cn, qzq@dlut.edu.cn

***Abstract*—In resource-limited environment, grid users compete for limited resources, and how to guarantee tasks' victorious probabilities is one of the most primary issues that a resource scheduling model cares. In order to guarantee higher task's victorious probabilities in grid resources scheduling situations, a novel model, namely ESPSA (Extended Second Price Sealed Auction), is proposed. The ESPSA model introduces an analyst entity, and designs analyst's prediction algorithm based on Hidden Markov Model (HMM). In ESPSA model, grid resources are sold through second price sealed auction. Moreover, to achieve high victorious probabilities, the user brokers who are qualified to participate in the auctions will predict other players' bids and then carry out the most beneficial bids. The ESPSA model is simulated based on GridSim toolkit. Simulation results show that the ESPSA model assures a higher victorious probability and superior to other traditional algorithms. Moreover, we analyze the existence of Nash equilibrium based on simulation results, thus, any participant who changes its strategy unilaterally could not make the results better.**

***Index Terms*—grid resource scheduling, game theory, extended second price sealed auction**

I. INTRODUCTION

The concept of grid technology originates from electricity grid. Idle resources in grids are connected through internet and grid users can access the resources no matter where they are. Grid system is heterogeneous, multi-zone managed, large-scale and distributed. Furthermore, grid system is concerned with a dynamic collection of diverse resources and services across multiple domains, and resources apply and demand is dynamic either. As resources join and exit dynamically, the loads of grid resources change continuously. All these characteristics show that grid system needs a dynamic and high-efficiency resources scheduling algorithm, which can manage dynamic resources flexibly.

Corresponding author: Zhenquan Qin

The majority of grid systems fall in the category of resource-rich ones, where resources compete for limited tasks. However, sometimes, numerous tasks are submitted at the same time, leading to the undersupplying of resources, which makes tasks compete for limited resources. Whether tasks compete for resources or resources compete for tasks, the resources providers will benefit from assigning resources and the thing that users need to do is paying for the resources. Thus, it is reasonable to model the resource scheduling by economic models. However, current literature on economic grid models only limits to the basic auction models where the bidders bid according to a single factor. Some algorithms integrate simple prediction methods into the bidding algorithms without assuring the victorious probability. Thus, with the increase of the complexity of grid systems, how to improve those simple bidding strategies, prediction algorithms and victorious probability is becoming urgent.

This paper is an extension in terms of both theoretic and simulation results of our conference paper [1]. The major contributions of this paper are as follows.

- To solve the task's low victorious probability problems using game theory in grid resource scheduling situations, an ESPSA (Extended Second Price Sealed Auction, ESPSA) model is proposed in this paper. An analyst entity is introduced in the model to fulfill the trend that roles in grids are increasing dramatically.
- The ESPSA model extends second price sealed auction mechanism. In the extended auction, the total price is sealed while keep the resource demand quantity unsealed.
- This paper introduces a prediction algorithm based on Hidden Markov Model, improves basic bidding algorithms and increases participants' victorious probability.
- GridSim toolkit is utilized to simulate the algorithms and the existence of Nash equilibrium is analyzed, and then proved theoretically.

The rest of paper is organized as follows: section II is the related work; section III describes the ESPSA model;

doi:10.4304/jcp.7.1.65-75

section IV elaborates the simulation and analyzes the simulation results and the computational load of the ESPSA, also the existence of Nash Equilibrium is proved theoretically in this part. Section V draws the conclusion and discusses about the future work.

II. Related Work

At present, resources management and dynamic scheduling based on game theory is becoming a focus of researches in the settings of P2P, grid and cloud computing. A lot of work has been done by researchers, and it will be overviewed in this section from the perspective of grid and cloud computing.

Game theoretic methods have been adopted in settings of cloud and grid computing. In the first place, from the grid computing perspective, Lei Yao and Guanzhong Dai (2008) [2] focus on resource-rich environment. They propose a method, namely GVP, which could guarantee resources' victorious probabilities. In their method, resources win the auction by predicting other resources' bids based on the mean value of historical bids. However, the prediction algorithm is too simple to get accurate predictions. Maheswaran and Basar(2003) [3] discuss a divisible auction and adopts a decentralized strategy, but users only make decisions according to resources' historical load information without consideration of the changing trend of resources' loads. Kwok and Song (2005) [4] propose a hierarchical grid model, in which resources are selfish. The selfishness of resources makes them only consider their own needs when executing tasks. This kind of action delays tasks' execution even though the tasks could accurately predict their opponents' bids. This model couldn't guarantee tasks' victorious probabilities. Bredin and Kotz (2003) [5] propose a resources allocation algorithm based on mobile brokers' competing for resources. In this model, user brokers make an inaccurate prediction for resources, and they neglect the prediction of other user brokers' behavior. LI Zhijie and Cheng Chuntian (2006) [6] propose a resource allocation model that uses sequential game to predict resource load for time optimization. In this model, user brokers offer their bids in sequence, causing first-mover advantage, which makes the model less popular in practical applications. Abramson and Buyya (2002) [7] propose grid resource scheduling model which only considers the relationship between users and resources without considering the interactions between users. Thus, this model is too idealized to be used in practical circumstances.

Game theoretic solutions also attracted a lot researchers' attention in the research area of cloud computing [8, 9, 10, 11, 12, 13, 14, 15]. Chonho, et al. (2010) [12] study an evolutionary game theoretic mechanism for adaptive and stable application deployment in cloud computing settings. Their algorithm, called Nuage, allows applications to adapt their locations and resource allocation to the environmental conditions in a cloud. Wei,G., et al. (2010) [13] consider a QoS constrained resource allocation problem, in which service demanders try to solve sophisticated parallel computing problem. Their solution consists of two steps. First, each participant solves its optimal problem independently, without consideration of the multiplexing of resource assignments. Then, an evolutionary mechanism which takes both optimization and fairness into account is designed. Rajkumar Buyya and Manzur Murshed (2002) [14] present a game theoretic method to schedule dependent computational cloud computing services and an evolutionary mechanism is designed. Fei and Frederic (2010) [15] introduce a new Bayesian Nash Equilibrium Allocation algorithm to solve resource management problem in cloud computing. Through experiments, they show that cloud users can receive Nash equilibrium allocation solutions by gambling stage by stage and the resource will converge to the optimal price.

The impact of selfish behavior entities has been paid a lot of attention in self-organized Mobile Ad hoc Networks (MANNEs) and Peer-to-Peer (P2P) networks [16, 17, 18, 19, 20, 23, 24, 25]. In such networks, each node specified as player is under the authority of a self-interested user.

All the literatures [2-25] listed above have made remarkable contributions to economic resources scheduling and allocation algorithm in the settings of grid and cloud computing, but some of them have a low accuracy prediction method for agents to predict tasks' bids. In response to these issues, a model, namely ESPSA, based on extended second price sealed auction and Hidden Markov Chain prediction method is proposed. In ESPSA model, a mass of tasks compete for limited resources, and tasks decide their best bidding price according to their resource demand quantities, their budgets and predictions of their opponents' bids. In this way, the ESPSA model will guarantee the tasks a higher victory probability when competing for grid and cloud resources than other methods do in the above literatures.

III. ESPSA Resource Allocation Model and Algorithm

The ESPSA model [1] is consisted of extended second price auction algorithm, resource broker's algorithm, user broker's algorithm with HMM. Specifically, the extended second price auction sealed the total price of each trade and unsealed the resource demand quantity. By invoking the extended auction, the users' and resources' brokers represent as bidders and auctioneers respectively.

In the common second price sealed auction [22], the bidders only submit prices that they wish to pay to the auctioneer without specifying a total price or a unit price. The ESPSA model extends the common second price sealed auction by specifying that the bidders should seal their total price and unseal their resource demand quantity. Then, after each auction, the auctioneer announces the winners and amount of resource they demand. This improvement is reasonable for that the resource broker can attract more customers (resources) by announcing the amount of resources processed in each auction. It is also applicable to seal customers' total price for privacy reason, because in some reality, the customers are not willing to let others know the total price they paid.

As Figure 1 shows, when a user wants to execute some tasks, it sends the task list to a user broker UB_i. After receiving the task list, in order to be qualified to take part in an auction, UB_i sends an application message with its expected victorious probability and amount of resource that it demands to resource broker RB_i. If resource broker RB_i agrees UB_i to take part in the auction, UB_i sends prediction request to an Analyst. On receiving the predictions, UB_i calculates its bid according to the prediction results. After that, it will send its bid to RB_i. Finally, RB_i sends winners to UB_i and resource demand quantity of UB_i to information service entity.

One of the improvements of our model is the analyst entity that we proposed. An analyst gets information from the information service entity, and provides prediction services to user brokers. The analyst entity's prediction algorithm is based on HMM. Obviously, it satisfies the trend that roles in grids are increasing dramatically and makes the market model of grids computing much more practical.

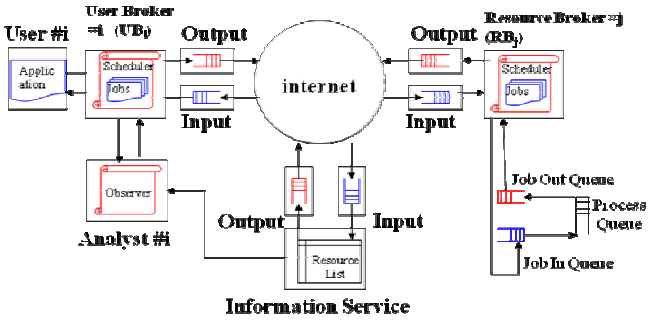


Figure 1. ESPSA model

The detailed user broker algorithm, resource broker algorithm and ESPSA prediction algorithm are described in section A, B and C respectively.

A. Algorithm of Resource Broker

As described above, ESPSA considers a resource-limited grid model where resource brokers need to act as auctioneers who invoke auctions among user brokers. For a resource broker, if it makes a higher expected profit for resources, it will be favored by the resources. Thus, the resource broker should limit the number of the bidders in order to fulfill most of the bidders' expected victorious probability in one round auction.

We propose a method which classifies the S_{bidder} into two sets: S^u_{bidder} and S^v_{bidder} represent users with an expected victory probability which is higher or lower than the basic victory probability set by the model respectively.

TABLE I.
RESOURCE BROKER'S ALGORITHM

Algorithm Name	Resource Broker's algorithm
Input:	A set of Bidders who expect to take part in the auction (S_{bidder}), bidders' expected victorious probability $\{\Pr_i^{win}\}$ and bidders' bids.
Output:	Messages that invoking a auction and winning messages.
1.Classify the bidders into S^u_{bidder} and S^v_{bidder} according to the user's task's average tasks'	
2.Invoke the first round auction where the bidders are in the set S^u_{bidder}.	
3.Waiting until received the bidding information $\{(b_{ij}, d_{ij})\}$ from user brokers.	
4.Calculate each user broker's the unit price by $r_{ij} = b_{ij} / d_{ij}$. Set the user broker with highest r_{ij} as the winner.	
5.Send need vector $D_j = \{d_{1j}, d_{2j}, \ldots d_{ij}, \ldots, d_{nj}\}$ to Information Service Entity (shown in Figure.1).	
6.If $\min(\Pr_i^{win}) > \Pr_{RB}^{win}$. Invoke a new auction where the bidders are in the set S^v_{bidder}.	
7.Else, rank S^v_{bidder} by the resource demand amount in increasing order. Add the top $1/\Pr_{RB}^{win}$ users to the new set S'_{bidder}. Invoke a new auction where the bidders are the new set S'_{bidder}.	
8.Waiting until received the bidding information $\{(b_{ij}, d_{ij})\}$ from user brokers.	
9.Calculate each user broker's the unit price by $r_{ij} = b_{ij} / d_{ij}$. Set the user broker with highest r_{ij} as the winner.	
10.If there were any request, then start another auction round. Else, exit.	

The number of bidders in each round should be calculated by Eq.(1).

$$n_{bidder} = \begin{cases} |S^u_{bidder}|, \text{in the 1st round auction and } \min(\Pr_i^{win}) > \Pr_{RB}^{win} \\ 1/\Pr_{RB}^{win}, \text{else cases in 1st round} \\ |S^v_{bidder}|, \text{in the 2nd round auction} \end{cases} \quad (1)$$

where $\Pr_i^{win}$ specifies the expected victorious probability of user broker UB_i, and $\Pr_{RB}^{win}$ represents the basic victorious probability that resource broker RB assures.

The specific algorithm of resource broker RB_i is described in Table I.

B. Algorithm of User Broker

User brokers need to represent users to join in the auction for resources. Apparently, user agents with higher victorious probability will get more users. The analyst entity provides predicting services. User brokers can

predict other user brokers' bids to get a high victorious probability by using the predicting services provided by an analyst.

Specifically, a user broker calculates its primitive bid by Eq.(2)[1].

$$b_i' = \gamma_i / L \tag{2}$$

Where b_i' is a user broker's primitive bid, L is the load status of the resource and γ_i is a weighting factor ($i = 1,2,\ldots,i,\ldots n$).

TABLE II.
USER BROKER'S ALGORITHM

Algorithm: User Broker's algorithm Input: A task set T from user. Output: Bid b_i
1. Firstly, user broker should apply to join in the auction. Send the applying request along with its expected victorious probability and deadline.
2. If the user broker is divided into the urgent one, then join auction, go to step 3. Else, wait for the second auction, go to step 3.
3.Calculate the primitive bid by Eq.(2)
4.Send predicting request to an analyst.
5.Waiting until received the predicting information B_{pre}^{-i} from the analyst.
6.Adjust the bid by Eq.(3).

Then, UB_i sends a predicting request to the analyst and gets a set of other user brokers' bids B_{pre}^{-i}. After that, UB_i adjusts its bid according to Eq.(3).

$$b_i = \begin{cases} \max\left(\{b_i'\} \cup B_{pre}^{-i}\right) + \varepsilon, & if \quad \alpha_i > \max\left(\{b_i'\} \cup B_{pre}^{-i}\right) \\ b_i' & else \end{cases} \tag{3}$$

Where α_i is a user broker's budget in one round of the auction. b_i is the calculated bid of a user broker.

The user agents' algorithm is listed in Table II.

C. ESPSA Model's Prediction Algorithm

In ESPSA model, the entity of analyst based on HMM prediction model is introduced. In our HMM prediction model, we use the Viterbi Algorithm[26].

In HMM prediction model, hidden states couldn't be got directly, but they can be deduced by observation sequences. In general, $\lambda = (A, B, \pi)$ can be used to represent a HMM, where A represents a transition probability between initial states, B represents observation sequence, π represents initial states matrix. Given the definitions of A, B and π, the HMM can generate a hidden sequence: $O = O_1, O_2, O_3, \ldots, O_n$.

In ESPSA model, the resource demand quantities and bids that are submitted by user brokers correspond to observation sequence and hidden states respectively. When an auction launched, each resource broker publishes its attributes, such as process elements' computing capacity, load status and so on. Each user broker can get the attributes. And then, user brokers use observed resource demand quantities $D_j = \{d_{1j}, d_{2j}, \ldots d_{ij}, \ldots, d_{nj}\}$ to predict their rivals' bids in next round auction: $b_j = \{b_{1j}, b_{2j}, \ldots b_{ij}, \ldots, b_{nj}\}$. d_{ij} is the demand quantity that user broker i of resource j, b_{ij} is the bid price of user broker i for resource j. We take five bids for examples, written as level1_bidding, level2_bidding, level3_bidding, level4_bidding and level5_bidding respectively, which correspond to the zones of $(0, b_1], (b_1, b_2], (b_2, b_3], (b_3, b_4], (b_4, b_5]$, where b_k is a bid, $k = 1,2,3,4,5$.

At time t, for user i, its bid for resource j is b_{ij}^t, and its resource demand quantity for j is d_{ij}^t. Suppose that there were n resources in grid system, according to HMM, the initial states at time t are:

$$\pi = states = \left(b_{i1}^t, b_{i2}^t, \ldots, b_{ij}^t, \ldots, b_{in}^t\right) \tag{4}$$

And at time t, the resource demand quantity matrix that user broker i observes other user brokers' is:

$$B = observations = \left(d_{i1}^t, d_{i2}^t, \ldots, d_{ii}^t, \ldots, d_{in}^t\right) \tag{5}$$

The transition matrix from hidden states at time $t-1$ to hidden states at time t is:

$$\mathrm{A} = \text{transition\_probability} = \begin{array}{c} \\ b_{i1}^{t-1} \\ \\ b_{in}^{t-1} \end{array} \begin{array}{c} \begin{array}{ccc} b_{i1}^t & \cdots & b_{in}^t \end{array} \\ \begin{pmatrix} p_{i11}^t & \cdots & p_{i1n}^t \\ \vdots & \ddots & \vdots \\ p_{in1}^t & \cdots & p_{inn}^t \end{pmatrix} \end{array} \tag{6}$$

The emission matrix between observation sequences and hidden states is:

$$emission\_probability = \begin{array}{c} \\ sta\_1 \\ \\ sta\_n \end{array} \begin{array}{c} \begin{array}{ccc} obs\_1 & & obs\_n \end{array} \\ \begin{pmatrix} p_{11}^t & \cdots & p_{1n}^t \\ \vdots & \ddots & \vdots \\ p_{n1}^t & \cdots & p_{nn}^t \end{pmatrix} \end{array} \tag{7}$$

where p_{ij}^t represents the probability of obs_j when in state sta_i at time t, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,n$.

As stated above, in ESPSA model, the hidden states are correlated with observation states by emission matrix. Specifically, given the hidden and observation states corresponding to bidding prices and resource demand quantity respectively, the ESPSA can predict the user brokers' bids.

IV. SIMULATION AND RESULTS ANALYSIS

The GridSim toolkit [23] is utilized in the simulation of this paper. Resource entities, user entities, broker entities and information about resource demand are considered.

A. Expriment Setup

It is necessary to make some initial setting reasonable by training especially the parameters used in HMM. There exist 300 resources in the resource queue, which means the user brokers may take part in at most 300 auctions. Table III lists the properties of the four resources.

TABLE III.
CHARACTERISTICS OF GRID RESOURCES IN GRIDSIM

Name	Resource characteristics	PE number	A PE MIPS rating	Transmitting Rate
R_0	Sun Ultra, Solaris	2	320	2000MB/s
R_1	Sun Ultra, Solaris	4	350	2000MB/s
R_2	Intel Pentium Linux	1	370	2500MB/s
R_3	Intel Pentium Linux	3	390	2500MB/s

Each gird user in this simulation has 2~ 4 tasks to execute on the gird and each task is with at least 2000MI (million instructions). In order to assure the variety of tasks, the size of each task is set to vary from -10%~+10% of 2000MI.

B. Simulation Methodology

The simulation process consists of two phases: parameter training, iterative rounds of auctions.

1) Parameter trainning. As the quality of the predictions depend on the model's parameters, these parameters have to be carefully chosen in order to maximize the prediction performance. We adopt Vertibi Training [26] which is an iterative training procedure that derives new parameter values from the observed counts of emissions and transitions in the Viterbi paths of the training sequences for the HMM parameters. In each iteration, a new set of parameter values is derived from the transitions and emissions in the sampled state paths for all sequences in the training set.

When HMM is initialized, the Viterbi segmentation is replaced by a uniform observation sequence. That is, the segmentation is divided into N equal parts. Each iteration, the means and variances are estimated. The training is stopped after a fixed number of iterations or as soon as the change in the log-likelihood is sufficiently small.

2) Equilibrium and winning probability. The winning probability is adopted to estimate the Nash equilibrium. Iterative auctions are simulated in repeated manner, and each round's winning probability is recorded for future analysis. When the winning probabilities at Nash Equilibrium are available, the results are analyzed by payoff matrixes in subsection D.

C. Results

We compare the HMM prediction model with random bidding model and average prediction model. And numerous results in predicting of bids, actual bids and victorious probability are stated in this section.

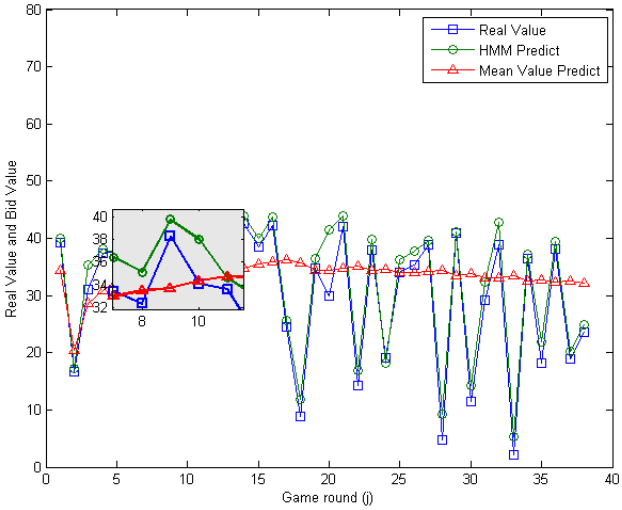


Figure 2. Comparison among real value, HMM prediction bids and Mean Value prediction bids

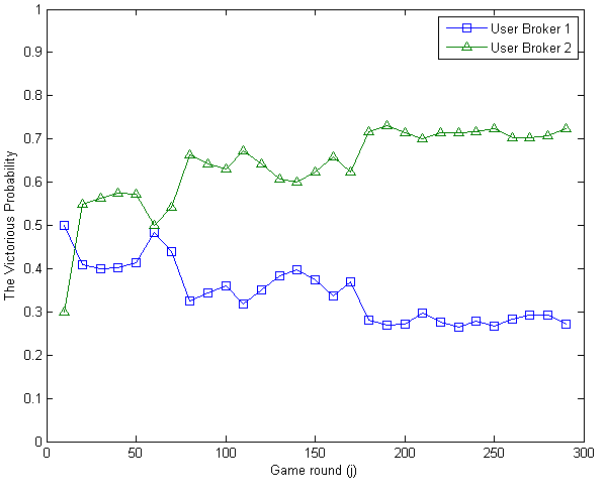
1) Comparison between prediction models: After the training of data, the maximum bid value is limited to 20. And the amount of resources is set randomly in [0,100]. After 40 rounds of auctions, from Figure 2, we can envisage the comparison between the actual bids and prediction bids. It is obvious that the trend of HMM prediction bids approximates the actual bids most of the time. However, the mean value prediction [2] bids fluctuate without any connection with the actual bids. Thus, we can conclude from the results that the prediction model based on HMM can approximate the actual bids better than the prediction method adopted by [2].

2) Victorious probability results of a two-player game. Suppose that two user brokers (UB_1 and UB_2) are competing in the game. Specifically, if UB_1 adopts no prediction algorithm and UB_2 utilizes the HMM prediction method, the results after 300 rounds of auctions are shown in Figure 3. It can be seen from Figure 3(a) that UB_2 can achieve a higher victorious probability.

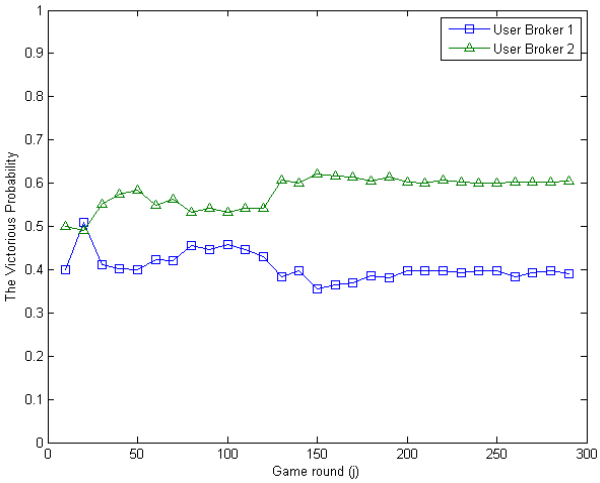
If UB_1 adopts the mean value prediction method, and UB_2 still uses the HMM prediction method, the results is depicted in Figure 3(b). It is apparent that UB_2 's victorious probability still exceeds the UB_1 's.

3) Victorious probability results of a multi-player game. On the analogy of the results of the two-player game, we can conclude that if all the players in the n-players game adopts the HMM prediction method, the victorious probabilities will converge at around 1/n. It means the expected victorious probabilities will be a very small value if a large amount of players join the auction.

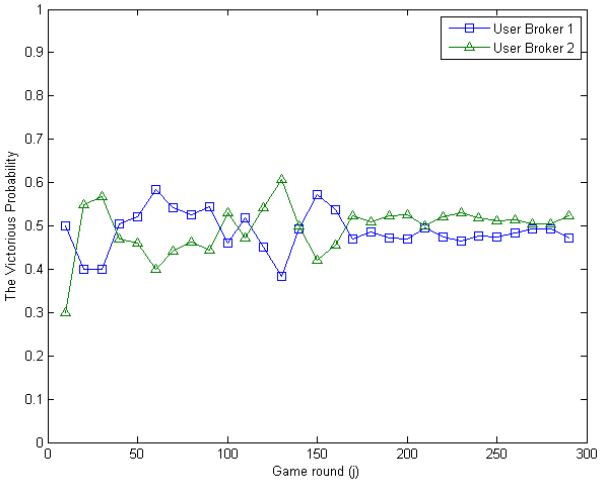
Thus, to avoid this problem, our algorithm of resource broker sets the number of bidders according to Eq.(1).



(a)



(b)



(c)

Figure 3. Victorious probabilities comparisons.(a) Random bidding vs. ESPSA model (b) Mean value prediction model vs. ESPSA model (c) ESPSA model vs. ESPSA model

Specifically, In ESPSA model, the method that we propose classifies the S_{bidder} into urgent set S^{u}_{bidder} and the less urgent set S^{v}_{bidder}. Then, according to each set, we propose two different rounds of auctions. The number of bidders in each round should be calculated by Eq.(1). So the resource broker limits the number of user brokers in each auction by this way and guarantees the high victorious probability.

In order to prove the superiority of the ESPSA algorithms with user broker number restrictions, we made a comparison among actual victorious probability, expected victorious probability under user broker number restrictions and expected victorious probability without user broker number restrictions. The comparison results are depicted as Fig.4. For easy reference, we use ESPSA-NNres to represent the ESPSA algorithms without user broker number restrictions. It can be seen from Figure.4 that the difference between expected victorious probability and actual victorious probability is much smaller in the ESPSA algorithm. In the 10*th* round of the auction, the average expected victorious probability is 0.4 and the result by ESPSA algorithm is 0.38. However, the ESPSA-NNres algorithm only gets a victorious probability of 0.34. Moreover, in the 30*th* round, user brokers' expected victorious probability is 0.58, the ESPSA's and ESPSA-NNres's victorious probabilities are 0.5 and 0.18 respectively.

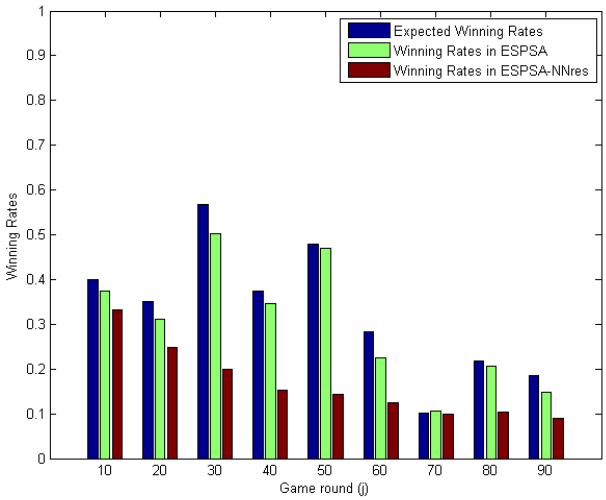


Figure 4. Comparison between expected and actual victorious probability

4) Resource utilization rate and execution time. By increasing the user brokers that take part in the GVP [2] and ESPSA game, the recorded resource utilization rate of the two algorithms are illustrated as Figure 5. It is worth to note that each data in Figure 5 is abtained through 200 rounds of auctions. From it, we can see that ESPSA in this paper can guarantee a higher resource utilization rate than GVP.

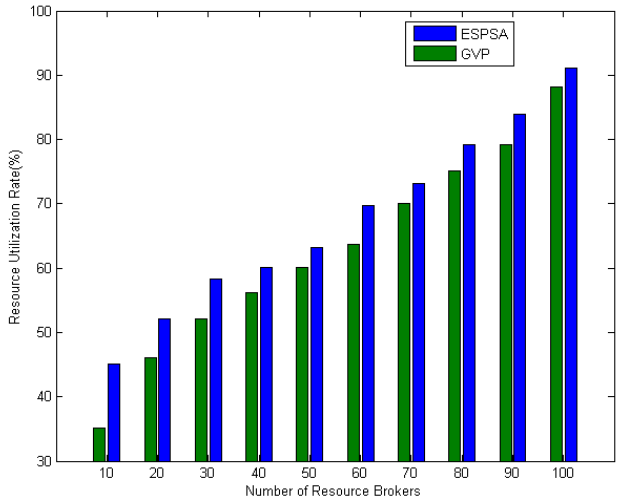


Figure 5. Resource Utilization Rate of ESPSA and GVP

From Figure 6, the algorithm execution time comparison between GVP and ESPSA algorithm is illustrated. It is obvious that the ESPSA algorithm runs more quickly at the most of circumstances.

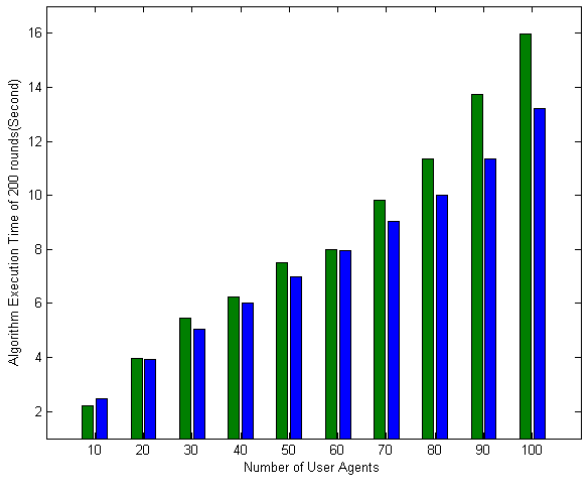


Figure 6. Algorithm execution time of ESPSA and GVP

Thus, according to the above analysis, we can see that the ESPSA model has higher superiority than GVP.

D. Equilibrium Analysis

Reference [2] utilized a mean value prediction method. Specifically, a player calculates his rival's bid in the current round of auction by Eq.(8),

$$\bar{b}^{j} = \frac{1}{j-1}\sum_{i=1}^{j-1} b^{i} \tag{8}$$

Then, the player only needs to bid a litter higher than the prediction bid.

We adopt the Cobb-Douglas production function to calculate the actual value (As Eq.(9) shows).

$$v(T,R,\alpha) = \eta\left((1-\alpha)\ln T + \alpha \ln R\right) \tag{9}$$

T represents the expected time that the user want to hold the resource. R represents the amount of resources that users want to have. α is a weighted factor which represents the preference of T and R. η is the profit that brought by the unit payoff. In the experiment, we set $\eta = 20$ and $\alpha = 0.95$. Given one user broker's historical resource demand quantities, through Eq.(9), the actual bid can be carried out.

By Eq.(8) and Eq.(9), the results are shown in Figure.2. The ESPSA algorithm performs better than the algorithm in [2] in approximating the actual value.

Figure.3 depicts the victorious probability results of random bidding, mean value prediction bidding and ESPSA prediction bidding strategies. Thus, the payoff matrixes can be given as follows.

		User Broker2: ESPSA Prediction Strategy	User Broker2: Random Prediction Strategy
User Broker1	ESPSA Prediction Strategy	0.5, 0.5	0.7, 0.3
	Random Prediction Strategy	0.3, 0.7	0.5, 0.5

Figure 7. Payoff matrix 1 of a two-user game

		User Broker2: ESPSA Prediction Strategy	User Broker2: Mean Value Prediction Strategy
User Broker1	ESPSA Prediction Strategy	0.5, 0.5	0.6, 0.4
	Mean Value Prediction Strategy	0.4, 0.6	0.5, 0.5

Figure 8. Payoff matrix 2 of a two-user game

From Figure 7 and Figure 8, it is obvious that the strict dominant strategy of user brokers is the ESPSA prediction strategy.

In a game, if a rational user agent has a strict dominant strategy, it will choose that strategy. In such way, the game reaches Nash equilibrium. For the reason that strict dominant strategy equilibrium must be Nash equilibrium [2], the two-player game above has Nash equilibrium:

$$S^{*} = (\text{ESPSA Strategy}, \text{ESPSA Strategy})$$

From the above analysis, we can conclude that the ESPSA predict strategy is superior to not only random bidding strategy but also mean value prediction strategy. Thus, all the rational players will choose the ESPSA strategy in the auction.

E. Computational Complexity of ESPSA

When weighing a model or algorithm is whether advanced or not, the computational complexity is an essential factor. In ESPSA model, the whole computational load depends on the HMM algorithm, that is the load of the analyst's calculation and prediction algorithm (Viterbi Algorithm).

The Viterbi Algorithm provides a computationally efficient way of analyzing observation of HMMs to recapture the most likely underlying state sequence [26]. It exploits recursion to reduce computational complexity and takes advantage of the context of the whole sequence to make judgments. In conclusion, using the Viterbi algorithm to decode an observation sequence (the resource demand quantities), the computational load will only be linear in T [26].

Thus, we can get that any users who participate the game using the ESPSA model will acquire the grid or cloud resources with high victorious probability within a very short time.

F. Proof of Nash Equilibrium in ESPSA

1) Nash Equilibrium of a Two-user Game

In this paper's setting, the users evaluate the value of an item according to the resource's load information. If we assume that each resource's load variation l is a normal distribution, the payoff function of each user can be given by Eq.(10).

$$u_i(b_1,\cdots,b_i,\cdots b_m;l)=\begin{cases}\gamma_i/l-b_i, & if\quad b_i\ \ wins\\ \dfrac{1}{m}\left(\sum_{i=1}^{m}\left(\gamma_i/l-b_i\right)\right), & if\quad b_1=\cdots=b_m\\ 0, & if\quad b_i\ \ loses\end{cases} \tag{10}$$

where γ_i is a constant, m is the total number of participants in the game, b_i is the bid price that user i competes for a resource, and $i=1,2,\ldots,m$. It is common a same load may bring different benefit for different users, so the constant γ_i varies according to each user's profile. L is the load vector.

Without loss of generality, initially, a game, in which two users (that is $m=2$) compete for one resource, is considered in the proof of Nash Equilibrium. Suppose that the strategies set of the users is $\{B_1,B_2\}$, and each user's bidding function is strictly increasing. If given user 1's bid, user 2's payoff function is

$$u_2(b_1,b_2;l)=\begin{cases}\gamma_2/l-b_2, & if\quad b_2>b_1\\ \dfrac{1}{2}\left(\sum_{i=1}^{2}\left(\gamma_2/l-b_2\right)\right), & if\quad b_1=b_2\\ 0, & if\quad b_2<b_1\end{cases} \tag{11}$$

To prove the existence of Nash Equilibrium, firstly, user 2's best response function need to be derived if given user 1's bid.

Lemma 1. Given user 1's bid strategy $b_1\in B_1$, user 2's best response is $b_2(l)=c_2\gamma_2/l$.

Proof. Suppose that user 2's probability density function is $f\left(\gamma/l\right)$ (for the ease of proof, we suppose $\eta_i=\gamma_i/l$, $i=1,2$), user 2's winning probability is

$$\begin{aligned}P_2^{win}(b)&=P\left(b_1(\eta_1)\le b\right)\\&=P\left(\eta_1\le b_1^{-1}(b)\right)\\&=\int_{\eta_1\le b}f(\eta_1)\,d\eta_1\end{aligned} \tag{12}$$

where b_1^{-1} is the reverse function of user 1's biding function. In order to win, user 2 has to find a optimal strategy in strategy set B_2. So, this problem can be modeled by a non-linear programming problem represented by Eq.(13).

$$\begin{aligned}&\max_{b_2}\int\eta_2 f_2(\eta_2)P_2^{win}\left(b_2(\eta_2)\right)d\eta_2\\&s.t.\\&\int\int_{\eta_1\le b}b_1(\eta_1)\,f_2(\eta_2)\,f(\eta_1)\,d\eta_1 d\eta_2\le\alpha_2\end{aligned} \tag{13}$$

where α_2 represents the budget of user broker 2 in a round of one auction.

It can be solved by a Lagrange multiplier. Then, we get Eq.(14).

$$\begin{aligned}&\int\int_{\eta_1\le b}\eta_2 f_2(\eta_2)\,f(\eta_1)\,d\eta_1 d\eta_2-\\&\lambda_2\left(\int\int_{\eta_1\le b}b_1(\eta_1)\,f_2(\eta_2)\,f(\eta_1)\,d\eta_1 d\eta_2-\alpha_2\right)\\&=\int\left[\int_{\eta_1\le b}\left(\eta_2-\lambda_2\cdot b_1(\eta_1)\right)f_1(\eta_1)\,d\eta_1\right]\cdot f_2(\eta_2)\,d\eta_2\end{aligned} \tag{14}$$

It is obvious that, if the Eq.(14) gets maximum value, the objective of Eq.(13) gets its optimal value. Therefore, when $\eta_2-\lambda_2\cdot b_1(\eta_1)=0$, the maximum value can be achieved. So, $b_{opt}=\eta_2/\lambda_2$. From $\eta_2=\gamma_2/l$, we get $b_{opt}=\gamma_2/l\lambda_2$. Suppose $c_2=1/\lambda_2$, it is easy to get $b_2(l)=c_2\gamma_2/l$. ■

Given user 2's best response, the existence of Nash Equilibrium can be derived.

Theorem 1. *In the ESPSA scheduling model with two users, there exists Nash Equilibrium.*

Proof: According to the Nash Existence Theorem [10, 11, 12], if each user's strategy space is non-empty, closed and bounded convex set in Euler space and the payoff function is continuous, quasi concave function, there exists a pure Nash Equilibrium.

From the assumptions of ESPSA model, the strategy sets B_1 and B_2 are non-empty, closed and bounded. According to Eq.(10), the payoff function is continuous in its domain. From Lemma 1, it is obvious that the best response, $b(l)=c_2\gamma_2/l$, is convex under its domain. Thus, payoff function is quasi concave [13]. ■

From the above prove, Nash Equilibrium exists in ESPSA model of two users.

2）Nash Equilibrium of N-winners Game

In the above section, we proved the existence of Nash equilibrium of two users for one auction. However, a

more realistic grid or cloud scheduling scenario setting is that multiple winners are allowed in one auction. It is because one resource always consists of several machines. Thus, in this section, we give a definition of the n-winner game and then propose a similar method of proving the Nash Equilibrium existence of a game.

Given user broker i's bidding strategy $b_i \in B_i$ and the strategy of user brokers except user broker UB_i is $b_{-i} \in B_{-i}$, $i = 1,2,\ldots,m$. User brokers aim to maximize their own expected benefit, while satisfying their budgets. Thus, the payoff function of user broker i can be written as Eq.(15).

$$u_i(b_i, b_{-i}; l) = \begin{cases} \gamma_i / l - b_i, b_i \in \max_n(b_1, b_2, \ldots, b_m) \\ 0, b_i \notin \max_n(b_1, b_2, \ldots, b_m) \end{cases} \quad (15)$$

where n is the number of winners, and we use $\max_n(b_1, b_2, \ldots, b_m)$, $0 < n \le m$, to represent the top n bids in $\{b_1, b_2, \ldots, b_m\}$.

Lemma 2. Given the payoff function of each user broker and the bidding strategy of other user brokers b_{-i}, in the n-winners ESPSA game, the best bidding strategy of a user broker i is in the form of $b_i = c\gamma_i / l$, where c is a constant.

Proof: The strategies for user broker i can be represented as the set of bidding functions $B_i = \{b_1, b_2, b_3 \cdots b_m\}$. In order to prove the existence of Nash Equilibrium, the form of response correspondence of each user broker $b(\cdot)$ has to be derived.

Given all the other user brokers' (except user broker i) bid b_{-i}, the expected payoff of user broker i is:

$$u_i = \left[\gamma_i / l - b_i\right] \cdot \Pr\left(b_i \in \max_n(b_1, b_2, \ldots, b_m)\right) \quad (16)$$

where $\Pr\left(b_i \in \max_n(b_1, b_2, \ldots, b_m)\right)$ is the probability of b_i in the set of top n bids. $\left(\gamma_i / l - b_i\right)$ represents the net payoff of UB_i when it wins the game. Let ω_n be the set of winners, and $\omega_{n_{-i}}$ be the n winners without the participation of UB_i. It is obvious that the bid of UB_i only needs to be greater than the least one in the set $\omega_{n_{-i}}$. Thus, we use $b_{Least(\omega_{-i})}$ to represent that user broker. Thus, to win the game, user broker i's bid only need to be greater than $b_{Least(\omega_{-i})}$. For the sake of simplicity, that user broker is represented by UB_L and its bid as b_L. Thus, UB_i's optimal bidding function can be represented by

$$\begin{aligned} &\max_{bi} u_i = \left[\gamma_i / l - b_i\right] \cdot \Pr\left(b_L < b_i\right) \\ &s.t. \\ &b_L \cdot \Pr\{b_L < b_2\} \le w_i \end{aligned} \quad (17)$$

ESPSA is an extended version of second sealed price sealed auction. It allows multiple winners and the payment of a winner k is the least winner's bid without the participation of the user broker k. Then, the payment of user broker i is b_L as the constraint term of Eq. (17) shows.

By applying the similar method, we can derive the result. The analysis is omitted for briefness. ■

Theorem 2: *A Nash Equilibrium exists in the ESPSA game* $G = \left[m, \{S_i\}, \{u_i(\cdot)\}\right]$ *with* $\{S_i\}$ *and* $u_i(\cdot)$ *defined above.*

Proof: The proof can follow the steps of Theorem 1 to show the existence of Nash equilibrium. For the sake of briefness, we omitted them here. ■

V. CONCLUSION AND FUTURE WORK

The ESPSA model is proposed in this paper and the model focuses on the resource allocation problem in grids and cloud computing. Firstly, we designed the interactions procedures between different entities in the ESPSA. Then, the algorithms of user and resource brokers are proposed. Specifically, in order to assure the victorious probability of each user broker, we introduce a bidder number restriction method in the resource broker's algorithm. Moreover, the user broker gets prediction information from the analyst who adopts the HMM prediction model. Then, based on the prediction results, the user broker carries out its bid. Simulation results show that, the ESPSA algorithms bids performs better in the approximation to actual bids than random bidding and mean value predict strategy. What's more, we analyzed the existence of Nash equilibrium based on the simulation results and the computational load of the ESPSA model. Lastly, we proved the existence of Nash equilibrium of the ESPSA model with two participants and multi-participant theoretically.

It is worthwhile to note that the ESPSA model is much more applicable than some basic auction models. However, the network delay, fraud user, reliability of resources problems are not considered in this paper. Thus, how to make the model realistic, fulfill the QoS requirements of users and improve the resource scheduling algorithms form the next step of our work.

ACKNOWLEDGMENT

This paper is partially supported by NSFC-JST under grant No.：51021140004; Nature Science Foundation of China under grant No.: 60673046, 90715037; University Doctor Subject Fund of Education ministry of China under grant No.: 200801410028; National 973 Plan of China under grant No.: 2007CB714205; Natural Science Foundation Project of Chongqing, CSTC under grant No.: 2007BA2024; Natural Science Foundation of China under Grant No. 60903153; the Fundamental Research Funds for the Central Universities; Nature Science Foundation of China under grant No.61070181.

REFERENCES

[1] Qiufen Xia, Weifeng Sun, Zichuan Xu, Mingchu Li, “A Novel Grid Resource Scheduling Model Based on Extended Second Price Sealed Auction”, 3rd International Symposium on Parallel Architectures, Algorithms and Programming, Dalian, pp. 305-310, 2010.
[2] Lei Yao, Guanzhong Dai, Huixiang Zhang, Shuai Ren, and Yun Niu, “A novel algorithm for task scheduling in grid computing based on game theory”, The 10th IEEE International Conference on High Performance Computing and Communications, pp. 282-287, 2008.
[3] Maheswaran, R.T., and Basar, T., “Nash equilibrium and decentralized negotiation in auctioning divisible resources”, Group Decision and Negotiation, pp. 361–395, 2003.
[4] Kwok, Y.K., Song, S.S. and Hwang, K., “Selfish grid computing: Game-theoretic modeling and NAS performance results”, In Proceedings of CCGrid, pp. 349–356, 2005.
[5] Bredin J., Kotz D., Rus D., Maheswaran R.T., Imer C., and Basar T., “Computational markets to regulate mobile-agent systems”, Autonomous Agents and Multi-Agent Systems, pp. 235–263, 2003.
[6] LI Zhijie, Cheng Chuntian, Huang Xuefei, and Li Xin, “A sequential game-based resource allocation strategy in grid environment”, Journal of Software, pp. 2373-2383, 2006. (in Chinese with English abstract).
[7] Abramson D., Buyya R., and Giddy J., “A computational economy for grid computing and its implementation in the nimrod-G resource broker”, Future Generation Computer Systems, pp. 1061−1074, 2002.
[8] Amazon Elastic Compute Cloud(EC2). http://aws.amazon.com/ec2.
[9] Microsoft Azure Services Platform. http://www.microsoft.com/azure/default. mspx.
[10] Rackspace Mosso. http://www.mosso.com/.
[11] Ristenpart, T., Tromer, E., Shacham, H., and Savage, S.. “Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds”, In Proceedings of the 16th ACM Conference on Computer and Communications Security, New York, pp. 199-212, 2009.
[12] Chonho Lee, Junichi Suzuki, Athanasios Vasilakos, Yuji Yamamoto, and Katsuya Oba., “An evolutionary game theoretic approach to adaptive and stable application deployment in clouds”, In Proceeding of the 2nd workshop on bio-inspired algorithms for distributed systems, ACM, New York, pp. 29-38, 2010.
[13] Guiyi Wei, Athanasios Vasilakos, Yao Zheng and Naixue Xiong, “A game-theoretic method of fair resource allocation for cloud computing services”, The Journal of Supercomputing, vol. 54, p. 252-269,2010.
[14] Rajkumar Buyya and Manzur Murshed, “GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing”, Concurrency and Computation: Practice and Experience, pp. 1175-1220, 2002.
[15] Fei Teng and Frédéric Magoulès, “A New Game Theoretical Resource Allocation Algorithm for Cloud Computing”, Advances in Grid and Pervasive Computing, pp. 321-330, 2010.
[16] Korilis. Y.A., Varvarigou. T.A. and Ahuja. S.R., "Incentive-compatible pricing strategies in non-cooperative networks", Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 439-446, 29 Mar-2 Apr 1998.
[17] Marbach, P., "Pricing differentiated services networks: bursty traffic”, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 650-658, 2001.
[18] Ranganathan. K., Ripeanu. M., Sarin. A. and Foster. I., "Incentive mechanisms for large collaborative resource sharing", Cluster Computing and the Grid, pp. 1- 8, April 2004.
[19] D.E. Volper, J.C. Oh and M. Jung, “GameMosix: Game-theoretic middleware for CPU sharing in untrusted P2P eEnvironment”, Parallel and Distributed Computing and Systems (SPDCS), 2004.
[20] Micah Adler, Rakesh Kumar, Keith Ross, Dan Rubenstein, David Turner, and David D. Yao. “Optimal peer selection in a free-market peer-resource economy”, In Proceedings of the Second Workshop on Economics of Peer-to-Peer Systems, 2004.
[21] Luzi Anderegg and Stephan Eidenbenz, “Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents”, In Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom '03), New York, pp. 245-259, 2003.
[22] Milgrom and Paul, "Auctions and Bidding: A Primer", Journal of Economic Perspectives, American Economic Association, pp. 3-22, 1989.
[23] Buyya R., Abramson D. and Giddy J., “A case for economy grid architecture for service-oriented grid computing”, In: Proceedings of the 10th IEEE Int’l Heterogeneous Computing Workshop, IEEE Computer Society, Washington, pp. 776−790, 2001.
[24] Guiyi Wei, Vasilakos. A.V. and Naixue Xiong, "Scheduling parallel cloud computing services: an evolutional game", 1st International Conference on Information Science and Engineering, pp. 376-379, December 2009.
[25] Durbin R, Eddy S, Krogh A and Mitchison G, “Biological sequence analysis: Probabilistic models of proteins and nucleic acids”, Cambridge, in Cambridge University Press, 1998.
[26] http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html

Weifeng Sun is an assistant professor in software school at the Dalian University of Technology(DLUT). Most of his research centers around high quality wireless communictions on wireless multihop network(such as WMN,WSN,MIPv6 and so on) and scheduling and applications on distributed network(including Grid Computing and Cloud Computing). He got the PH.D degree and bachelor degree at University of Science and Technology of China(USTC).Additional information about Sven can be found on his webpages:home.ustc.edu.cn/~wfsun.

Qiufen Xia, born in 1987, received the M.A. from Dalian University of Technology, Dalian in 2009. She is a master student of software school, Dalian University of Technology. Her study interests include grid resources scheduling, game theory, network security and cloud computing.

Zichuan Xu, born in 1986, male, received Master degree and B.E degree (all with highest honor) from Dalian University of

Technology, China. His research interests include approximation algorithms, temperature-aware scheduling, cyber-physical systems (CPS), game theory, green computing.

Mingchu Li received the B.S. degree in mathematics, Jiangxi Normal University and the M.S. degree in applied science , University of Science and Technology in 1983 and 1989, respectively. He worked for University of Science and Technology in the capacity of associate professor from 1989 to 1994. He received his doctorate in Mathematics, University of Toronto in 1997. He was engaged in research and development on information security at Longview Solution Inc, Compuware Inc. from 1997 to 2002. From 2002, he worked for School of Software of Tianjin University as a full professor, and from 2004 to now, he worked for School of Software of Dalian University of Technology as a full Professor, Ph.D. supervise, vice dean. His main research interests include theoretical computer science and cryptography.

Zhenquan Qin received the bachelor degree in security engineering and the Ph.D. degree in communication and informatin system both from University of Science and Technology of China in 2002 and 2007,respectively. Now he is a lecturer in the School of Software of Dalian University of Technology. His research interests include network analysis, network security and quantum communication.

A Performance Model for Network-on-Chip Wormhole Routers

Youhui Zhang
Department of Computer Science and Technology, Tsinghua University, Beijng, 100084, China
Email:zyh02@tsinghua.edu.cn

Xiaoguo Dong and Siqing Gan
Department of Mathematical Science and Computing Technology, Central South University, Changsha, 410075, China

Weimin Zheng
Department of Computer Science and Technology, Tsinghua University, Beijng, 100084, China

***Abstract*—A generic analytical performance model of single-channel wormhole routers is presented using the M/D/1/B queuing theory. Compared with previous work, the flow-control feedback mechanism is studied in detail, and a computing method bases on Markov chain for the flow-control feedback probability is proposed. Compared with BookSim, a well-known cycle-accurate Network-on-Chip (NoC) simulator, this model presents accurate results on key metrics: the average relative error of flow-control feedback probability is about 7.87%. In addition, based on the model of single-channel routers, the asymmetric multi-channel and symmetric multi-channel structured routers are both modeled respectively.**

***Index Terms*—Network-on-Chip; Markov chain; queuing theory; performance analysis**

I. INTRODUCTION

Networks-on-Chip (NoC) [1][2] has been proposed as a solution for addressing the design challenges of high performance nano-scale architectures by separating the on-chip communication from computing and storage. Connecting components through an on-chip network has several advantages over dedicated wiring, potentially delivering high-bandwidth, low-latency, and low-power communication over a flexible, modular medium.

Wormhole-routing [3][4] is a system of simple flow control in NoC based on fixed links, which makes message latency almost independent of the inter-node distance in the absence of blocking. In wormhole routing, network packets are broken into small pieces called flits. The first flit, called the header flit holds information about this packet's route (namely the destination address) and sets up the routing behavior for all subsequent flits associated with the packet. The head flit is followed by zero or more body flits, containing the actual pay load of data. The final flit, called the tail flit, performs some bookkeeping to close the connection between the two nodes. If the header is blocked, the data flits are blocked behind it occupying all the channels and buffers already taken [3].

Moreover, the routing algorithm defines how to transfer a message through a network path. A key issue for any routing algorithm is deadlock-free. Now deterministic routing for deadlock prevention is widely employed [3].

Currently, many NoC designs are based on the simulation method. However, simulation is a time-consuming procedure, especially within the large design space. Therefore, system designers have to choose limited assessment in the design space and then cannot get the optimized results usually.

Another approach is utilization of an analytical model of the system which is the most cost-effective tool for performance evaluation. Several analytical models of deterministic routing in wormhole-routed mesh-based networks, e.g. hypercube and tori, have been reported.

Ref.[5] introduces a probability model for wormhole network. Ref.[6] is restricted to the k-ary n-cubes topology while [7] is only adapted by the hypercube. Ref.[8] presents a performance model based on some queuing theory, but it only can apply to the switched network. Ref.[9] presents an analytical performance evaluation method for the general wormhole-routing NoC architectures. It analyses different reasons for blocking network packets in the router buffer, but does not consider the flow-control feedback mechanism, which is caused by the fullness of the input queue of the successive router. Ref.[10], based on the same assumptions of [9], provides a novel method: the numerical analysis and iterative computation is used to estimate the NoC performance.

Moreover, several models for adaptive wormhole routing have been introduced for the torus and hypercube [11][12][13].

This paper presents a generic analytical performance evaluation approach of NoC design. Different from the previous work, the flow-control feedback probability between adjacent routers is considered meticulously, which is the important indicator of the buffer utilization

Supported by Chinese National Science Foundation under Grant 60773147, and Chinese National 863 High Technology Programs under Grant 2008AA01A204, Email: zyh02@tsinghua.edu.cn.

doi:10.4304/jcp.7.1.76-84

and network traffic. And how to extend the single-channel model to the model for multi-channel routers is also presented.

In summary, this paper gives the following contributions:

1) A general analytical model of wormhole routers with single channel is proposed, which supports arbitrary network topologies, the deterministic routing algorithm, arbitrary packet / buffer lengths, and so on.

Based on the router model, the computing method of flow-control feedback probability using the Markov chain is presented in detail.

2) Based on the performance model of the single-channel router, the asymmetric multi-channel and symmetric multi-channel structured routers are both modeled respectively. Multi-channel structures can improve the communication performance significantly.

3) The accuracy of this approach is validated through the comparisons with a well-known cycle-accurate simulator, BookSim [14].

The remaining of this paper is organized as follows. Section 2 introduces the related work on performance modeling. Section 3 gives our modeling assumptions. The model for single channel routers is presented in the next section and the multi-channel versions are presented in Section 5. Experimental results are given in Section 6; the conclusion and future work are introduced at last.

II. Related Work

Ref. [5] develops a model of a single wormhole router, which is based on the probability analysis rather than queuing theory. The model is evaluated through a series of flit-level simulations. Moreover, how to extend the model to networks of routers is also discussed.

Ref. [6] analyzes communication networks of varying dimension under the assumption of constant wire bisection. Models of the latency, average throughput, and hot-spot throughput of k-ary n-cube networks are presented.

Ref. [7] proposes a general analytical model to predict message latency in wormhole-routed k-ary n-cubes with fully adaptive routing. The analysis focuses on a widely-accepted fully adaptive routing algorithm.

A system-level buffer planning algorithm is given in [8]. Using this algorithm, the buffer depth for each input channel in different routers across the chip can be derived to optimize the overall performance, given the traffic characteristics of the target application and the total budget of the available buffering space.

Ref. [9] presents a generalized router model and then utilizes this model for doing NoC performance analysis. The proposed model can be used not only to obtain fast and accurate performance estimates, but also to guide the NoC design process within an optimization loop.

Ref. [10] also presents a generic analytical method to estimate communication latencies and link-buffer utilizations for a given NoC architecture. The accuracy of this method is experimentally compared with the results obtained from Cycle-Accurate SystemC simulations. It is based on the same assumptions of [9], providing a novel method: the numerical analysis and iterative computation is used to estimate the NoC performance.

III. Assumptions of Router Modeling

A. NoC Router

An illustration of the router structure is presented in Fig.1.

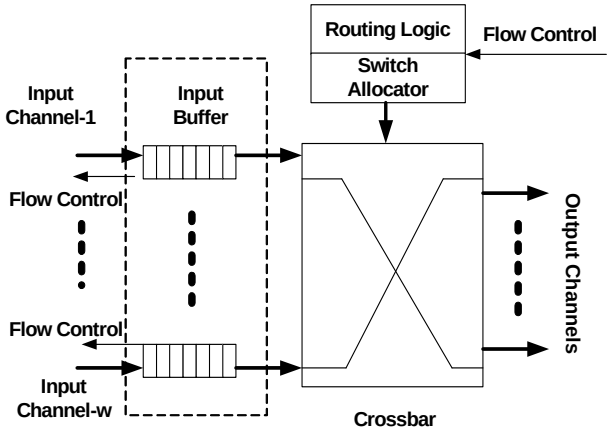


Figure 1. The structure of a router

We assume that a wormhole router contains w ports and adopts a deterministic routing algorithm (like X-Y routing). The local port is regarded as the same as others; each port is associated with a single input buffer (single channel; how to model the multi-channel structure is discussed in Section V). Input buffers are abstracted as groups of FIFO.

The crossbar switch can be configured to connect any input buffer of the router to any output channel, but under the constraints that each input is connected to at most one output, and each output is connected to at most one input. The tasks of resolving all the potential requests to the crossbar and other shared resources of the router fall onto the switch allocator.

Before transfer, data packets are divided into small pieces, called flits. The header flit holds destination information to set up the transfer channel for all subsequent flits of the same packet.

The router also implements flit-level flow control: the

TABLE I.
Parameters of the Model

Symbol	Description
w	Number of ports of a router
B	Length of the input buffer
T	Service time of a packet, which does not include the waiting time in the queue
P	Packet size (in flit)
H_s	Service time of the header flit, or the time of the header flit going through the router (without the waiting time). It is also the number of pipeline stages of a router because of Assumption 4.
(i, j)	Port j ($0 <= j < w$) at Router i.

input queue capacity is limited, so if any input queue of a router is full, its upstream router will stop transmitting. In the previous work, flow-control feedback is less involved or even ignored.

For the detailed workflow of a wormhole router, please refer to [15].

B. Modeling Parameters

As in [9][10], the router model introduces the following hypotheses.

1) Network traffic is generated from all nodes uniformly and follows the Poisson process, which implies that packet-arrival events occur continuously and independently of one another.

2) Packet destinations are equally distributed across the network nodes following a uniform traffic pattern.

3) Traffic sinks consume the incoming packets with the constant rate of one flit per cycle.

4) A pipeline stage of a router can deal with one flit per cycle, too.

5) Input buffers of any router have finite capacity.

6) When the system achieves asymptotic stability, the service time of packets is approximately equal to a constant value.

Some parameters of the model are listed in Table 1.

Then, we have (1) because of Assumption 4.

$$T = H_s + P \tag{1}$$

IV. Analytical Model of Single-Channel Routers

A. Analysis of waiting time in the input queue

This section focuses on modeling a single-channel router as a set of first-come first-serve buffers connected by a crossbar switch to analyze the average waiting time that an incoming packet spends in the queue.

$T_{i,j}$ denotes the average time in the queue of (i, j). It is composed of the following three parts as described in [9]:

1) Service time of the packets already waiting in the same buffer;

2) The residual service time seen by an incoming packet;

3) The packets waiting in other buffers of the same router and served before the incoming packet.

As stated in [9], the traffic rate at (i, j), $\lambda_{i,j}$, can be computed by (2).

$$\lambda_{i,j} = \sum_{\forall s}\sum_{\forall d} x_{s,d} R(s,d,i,j) \tag{2}$$

Here $x_{s,d}$ represents the traffic rate from source router s to the destination d, and $\pi_{s,d}$ indicates the path from s to d. R is the indicator function that returns 1 if the path goes through (i, j), and returns 0 otherwise.

$$R(s,d,i,j) = \begin{matrix} 1, (i,j) \in \pi_{s,d} \\ 0, (i,j) \in \pi_{s,d} \end{matrix} \tag{3}$$

I. The first part

Suppose the average number of packets in the input buffers of router i is a vector, *N*.

$$N = \left[N_1, N_2, ..., N_w \right]^T . \tag{4}$$

Then, the average number of packets waiting in port j is N_j . And the average waiting time for incoming packet is $E(T) \times N_j$, where $E(T)$ indicates the mean of service time.

II. The second part

If incoming packet p_m arrives at the top of the input queue while some other packet p_n is being serviced, then the residual service time R_m for p_m is the time left for the packet p_n to finish its service.

To simplify the analysis, the concept of mean residual service time $R(\lambda_{i,j})$ is used. It is the average value of service time for all packets, as well as a function of the traffic rate and service time. When the system reaches the asymptotic steady-state, the following equation represents its value:

$$R(\lambda_{i,j}) = \frac{1}{2}\lambda_{i,j} \times E(T^2) . \tag{5}$$

where $E(T^2)$ is the second moment of service time.

III. The third part

In [9][10], both Part 1 and 2 have been analyzed completely. But for Part 3, they do not consider the flow-control feedback. Therefore, we focus on this issue.

In detail, when a head flit intends to go to the specific output port, it has to compete with all other flits applying for the same direction. Moreover, another necessary condition for any winner to continue is that the input queue of the downstream router is not full, which is called the flow-control feedback.

Then, a packet transmitted from (i, j) to $(i+1, k)$ consists of two processes: *competition* and *flow-control*.

Suppose $F_{i,j,k}$ is the probability of the header flit transmitted from (i, j) to $(i+1, k)$, and $p_{i+1,k}$ is the flow-control feedback probability produced from $(i+1, k)$ and $f_{i,j,k}$ is the competition probability of the header flit. Then we have:

$$F_{i,j,k} = f_{i,j,k} \times (1 - p_{i+1,k}) . \tag{6}$$

$\lambda_{i,j,k}$ is the traffic rate from (i, j) to $(i+1, k)$ and we get

$$f_{i,j,k} = \frac{\lambda_{i,j,k}}{\sum_{l=1}^{w} \lambda_{i,l,k}} . \tag{7}$$

$c_{i,j,q}$ denotes the competition probability of the header flits in (i, j) and (i, q) transmitting to the same input port of Router *(i+1).*

We have $c_{i,j,q}$ = 1 if j=q.

If $1 \le j,q \le p$ and $j \ne q$, we can get

$$c_{i,j,q} = \sum_{k=1}^{w} F_{i,j,k}F_{i,q,k} = \sum_{k=1}^{w} f_{i,j,k}\, f_{i,q,k}(1-p_{i+1,k})^2 \quad (8)$$

Therefore, the blocking delay caused by packet competitions and flow controls can be denoted by (9).

$$E(T)\sum_{q=1,q\ne j}^{w} c_{i,j,q}N_q = E(T)\sum_{q=1,q\ne j}^{w}\sum_{k=1}^{w} f_{i,j,k}\, f_{i,q,k}(1-p_{i+1,k})^2 N_q . \quad (9)$$

Summing up the three parts, we obtain the average waiting time of an incoming packet buffered in (i, j).

$$T_{i,j} = E(T)N_j + \frac{1}{2}\lambda_{i,j}E(T^2) + E(T)\sum_{q=1,q\ne j}^{w}\sum_{k=1}^{w} f_{i,j,k}\, f_{i,q,k}(1-p_{i+1,k})^2 N_q \quad (10)$$

To calculate $T_{i,j}$, it is necessary to computer the flow-control feedback probability, $p_{i+1,k}$.

B. A computing method of flow-control feedback probability

When the system arrives at the asymptotic steady-state, the service time of packets can be regarded as the mean value, $E(T)$. In general, we consider the flow-control feedback probability $p_{i+1,k}$ of the input queue at (i+1, k). In this figure, $p_{i+1,k}$ is produced by (i+1, k) with the arrival rate $\sum_{j=1}^{w} f_{i,j,k}\lambda_{i,j,k}$ and the service rate $\frac{1}{E(T)}$.

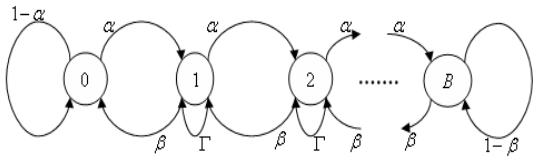


Figure 2. State transition diagram for M/D/1/B queue

Using Markov chain to analyze the changes of flits in the input queue, the state transition diagram for the queue is shown in Fig.2 and the state transition matrix can be written as follows.

$$M = \begin{bmatrix} 1-\alpha & \alpha & 0 & \cdots & 0 & 0 & 0 \\ \beta & \Gamma & \alpha & \cdots & 0 & 0 & 0 \\ 0 & \beta & \Gamma & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \Gamma & \alpha & 0 \\ 0 & 0 & 0 & \cdots & \beta & \Gamma & \alpha \\ 0 & 0 & 0 & \cdots & 0 & \beta & 1-\beta \end{bmatrix} \quad (11)$$

where

$$\alpha = (\sum_{j=1}^{w} f_{i,j,k}\lambda_{i,j,k}) \times (1-\frac{1}{E(T)}) ; \quad (12)$$

$$\beta = (1-\sum_{j=1}^{w} f_{i,j,k}\lambda_{i,j,k}) \times \frac{1}{E(T)} ; \quad (13)$$

$$\tau = (\sum_{j=1}^{w} f_{i,j,k}\lambda_{i,j,k}) \times \frac{1}{E(T)} + (1-\sum_{j=1}^{w} f_{i,j,k}\lambda_{i,j,k}) \times (1-\frac{1}{E(T)}) ; \quad (14)$$

According to the state transition diagram, we get the equilibrium distribution vector,

$$S_{i+1,k} = \left[S_{i+1,k,0}, S_{i+1,k,1}, \ldots, S_{i+1,k,B}\right]^T , \quad (15)$$

where $\sum_{n=1}^{B} S_{i+1,k,n} = 1$ (16)

In (15), $S_{i+1,k,n}$ is the probability of the state having n flits filled in the input queue of $(i+1,k)$ and $S_{i+1,k,0}$ is the probability of an empty queue; $S_{i+1,k,B}$ is the probability of a full queue, which can be called the probability generating the flow-control feedback from (i+1, k).

The difference equations for the state transition distribution vector can be written as follows.

$$\alpha S_{i+1,k,0} - \beta S_{i+1,k,1} = 0 \quad (17)$$

$$\alpha S_{i+1,k,n-1} - (\alpha+\beta)S_{i+1,k,n} + \beta S_{i+1,k,n+1} = 0 \quad (0 < n < B) \quad (18)$$

Then, the solution of the above difference equations can be gotten as

$$S_{i+1,k,n} = (\frac{\alpha}{\beta})^n S_{i+1,k,0} (0 \le n \le B) \quad (19)$$

We define the duty factor of the system as

$$\rho = \frac{\alpha}{\beta} = \frac{(\sum_{j=1}^{w} f_{i,j,k}\lambda_{i,j,k})(1-\frac{1}{E(T)})}{(1-\sum_{j=1}^{w} f_{i,j,k}\lambda_{i,j,k})\frac{1}{E(T)}} \quad (20)$$

where

$$\sum_{n=0}^{B} S_{i+1,k,n} = S_{i+1,k,0}\sum_{n=0}^{B}\rho^n = 1 \quad (21)$$

Now, we get

$$S_{i+1,k,0} = \frac{1-\rho}{1-\rho^{B+1}} \quad (22)$$

And then we have

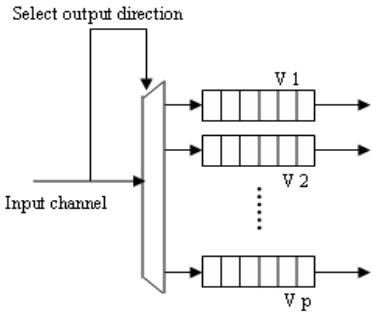
$$p_{i+1,k} = S_{i+1,k,B} = \rho^B \frac{1-\rho}{1-\rho^{B+1}} (1 \le k \le p) . \quad (23)$$

V. ANALYTICAL MODEL FOR MULTI-CHANNEL STRUCTURES

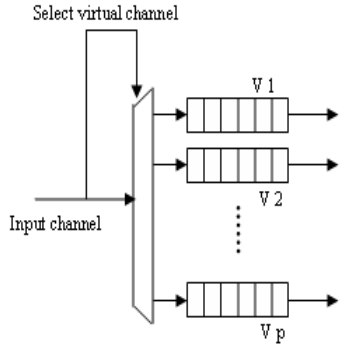
As the increase of network traffic, the waiting time of packets in the input queue of single-channel routers increases, which results in longer transmission delay. To solve this problem, there are two ways: reduce the packet arrival rate and improve the service rate. At present, NoC routers usually adapt multi-channel architecture to achieve the two functions.

Ref. [16] has classified multi-channel routers into two categories: the asymmetric multi-channel and symmetric multi-channel structures. Both are modeled based on our above-mentioned work. In [16], the flow-control feedback possibility is computed based on the M/G/1 queue theory, while we calculate the possibility based on the M/D/1/B queue theory and the Markov chain.

A. Two typical multi-channel NoC routers



(a) asymmetric multi-channel structure



(b) asymmetric multi-channel structure

Figure 3. Multi-channel structure NoC rouetrs

Fig.3(a) shows the asymmetric multi-channel structure: each input port is associated with *p* channels which match different output ports respectively; in another word, each channel matches only one output port and different channels match different output ports respectively. Flits select output direction according to their own routing information. Then the traffic is divided into different channel queues for transmission, thus the arrival rate for each buffer is reduced and the service rate is also improved.

Fig.3(b) shows the symmetric multi-channel structure: each input port is also associated with *p* channels, and each has the same function. The incoming packet select virtual channel with the same probability 1/p. Different from the asymmetric, here each channel can match any one of output ports.

In the symmetric structure, the traffic is divided into several groups with same probability, and then each channel send the transmission request to the corresponding output port. If the current packet fails in the competition and is being blocked, the subsequent packets to the same output port can enter the other channels for transmission. The structure can also reduce the arrive rate and improve the service rate.

B. Analytical model for multi-channel structure

I. Analytical model for asymmetric multi-channel structure

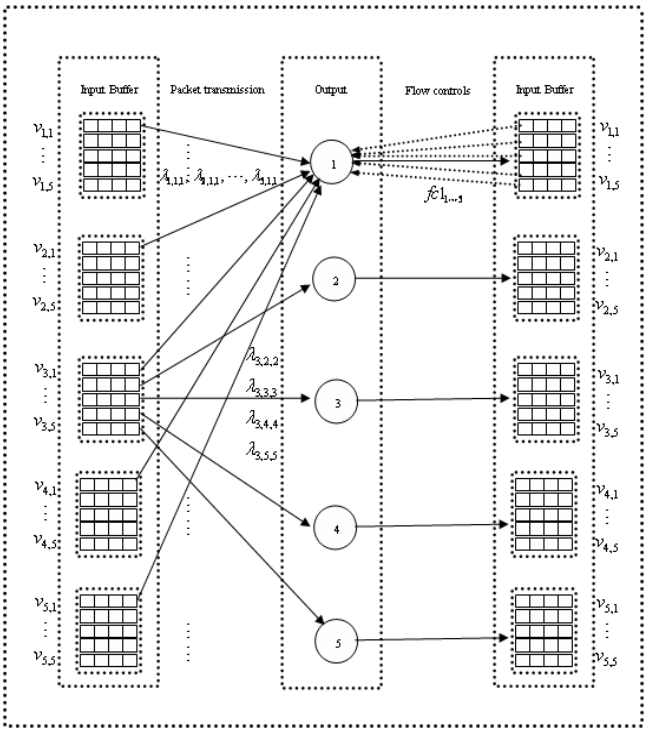


Figure 4. Arbitration model of asymmetric multi-channel structure

Assuming each input port of an asymmetric multi-channel router is associated with *p* channels. For example, for the 2D-mesh topology, each input port has five channels and each is connected to one of the five different output directions; the traffic transmission is illustrated in Fig.4. In this structure, the flits in the queue only compete with the flits with the same transfer direction from other input ports. For example, the flits in channel $V_{2,1}$ only compete with the flits from channel $V_{1,1}$, $V_{3,1}$, $V_{4,1}$, and $V_{5,1}$. Different from the single-channel structure, the flow-control occurs only if all five channels of the corresponding downstream router are fulfilled.

$\lambda_{i,j,k,k}$ represents the traffic rate from the channel k ($1 \le k \le p$) of (i, j), which can be computed by (24).

$$\lambda_{i,j,k,k} = \sum_{\forall s} \sum_{\forall d} x_{s,d} \times R(s,d,i,j,k) \times S(s,d,k,k) \quad (24)$$

Similar with (2), here R is the indicator function that returns 1 if the path goes through the k channel of (i, j), and returns 0 otherwise. Function $S(s,d,k,k)$ returns 1

if and only if the flits in the channel *k* will be sent to output port *k*, otherwise it returns 0. Because the flits in each input channel *k* can only be sent to the output port *k*, we have $S(s,d,k,k)=1$.

The analysis of waiting time in the input queue k of (i, j) also includes three parts. The first and the second parts are similar with the single-channel structure. We focus on the third.

Suppose $F_{i,j,k,k}$ is the probability of the header flit transmitted from the channel k in (i,j) to $(i+1,k)$, $p_{i+1,k}$ is the flow-control feedback probability produced from $(i+1,k)$ and $f_{i,j,k,k}$ is the competition probability of the header flit. We have

$$F_{i,j,k,k} = f_{i,j,k,k} \times (1 - p_{i+1,k}^{p}) \tag{25}$$

$$f_{i,j,k,k} = \lambda_{i,j,k,k} \Big/ \sum_{l=1}^{p} \lambda_{i,j,l,k} \tag{26}$$

$c_{i,j,q,k}$ denotes the competition probability of the header flits in the channel k of (i, j) and (i, q) transmitting to the same input port of router *(i+1)*.

We have $c_{i,j,q,k}$ = 1 if j=q.

If $1 \le j,q \le p$ and $j \ne q$, we can get

$$\begin{aligned} c_{i,j,q,k} &= F_{i,j,k,k} \cdot F_{i,q,k,k} \\ &= \begin{cases} 1 & (j=q) \\ f_{i,j,k,k} f_{i,q,k,k} (1 - p_{i+1,k}^{p})^2 & (1 \le j,q \le p; j \ne p) \end{cases} \end{aligned} \tag{27}$$

Therefore, the blocking delay caused by packet competitions and flow controls happened in the two channels of (i, j) and (i, q) can be denoted by (28).

$$\begin{aligned} & E(T) \sum_{q=1,q\ne j}^{p} c_{i,j,q,k} N_{q,k} \\ & = E(T) \sum_{q=1,q\ne j}^{p} f_{i,j,k,k} f_{i,q,k,k} (1 - p_{i+1,k}^{p})^2 N_{q,k} \end{aligned} \tag{28}$$

Summing up the three parts, the average waiting time of an incoming packet buffered in channel k of (i, j) is:

$$\begin{aligned} \tau_{i,j,k} &= E(T) N_{j,k} + \frac{1}{2} \lambda_{i,j,k} E(T^2) \\ &+ E(T) \sum_{q=1,q\ne j}^{p} f_{i,j,k,k} f_{i,q,k,k} (1 - p_{i+1,k}^{p})^2 N_{q,k} \end{aligned} \tag{29}$$

where $N_{q,k}$ $(1 \le q,k \le p)$ stands for the average number of flits in the channel k of (i, q).

II. Analytical model for symmetric multi-channel structure

Routers can also adopt symmetric multi-channel structure. Each input port is associated with *p* channels. The incoming packet selects any virtual channel with the same probability, 1/p. Still taking the 2D-mesh topology as the example, each input port has five channels, the flits select a virtual channel with the probability of 0.2, and each channel can send flits to arbitrary output port. It means that any output port can receive the flits from all 25 channels.

The traffic transmission is shown in Fig.5. Compared with the asymmetric structure, the transfer principle is the same, but the flow-control probability is different.

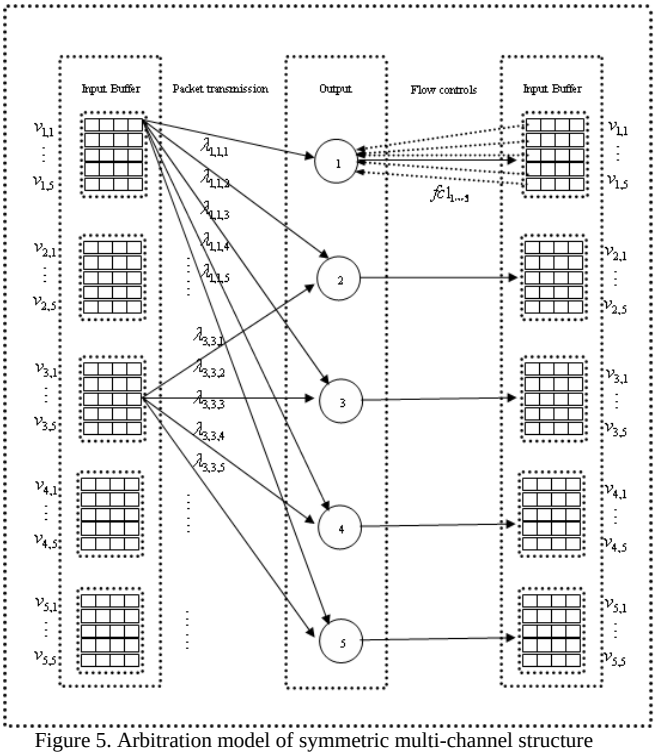


Figure 5. Arbitration model of symmetric multi-channel structure

$\lambda_{i,j,k,h}$ presents the traffic rate from channel k of (i,j) to output port $h(1 \le h \le p)$, then we have

$$\lambda_{i,j,k,h} = \sum_{\forall s} \sum_{\forall d} x_{s,d} \times \frac{R(s,d,i,j,k) S(s,d,k,h)}{p} \tag{30}$$

Here, R is the indicator function owing the same meaning of the asymmetric version. Function S returns 1 if and only if the flits in the channel *k* will be sent to output port *h*, otherwise it returns 0.

The analysis of waiting time in the input queue k of (i, j) still includes three parts. The first and second parts are the same with the asymmetric multi-channel structure. We also focus on the third part and we have

$$F_{i,j,k,h} = f_{i,j,k,h} \times (1 - p_{i+1,h}^{p}) \tag{31}$$

$$f_{i,j,k,h} = \lambda_{i,j,k,h} \Big/ \sum_{j=1}^{p} \sum_{l=1}^{p} \lambda_{i,j,l,h} \tag{32}$$

Here F, f, and p have the same meanings of the asymmetric model.

Similarly, $c_{i,j,q,k}$ denotes the competition probability of the header flits in the channel *k* of (i, j) and (i, q) transmitting to the same input port of Router (i+1).

If $j = q$, we have

$$c_{i,j,q,k,n} = \sum_{h=1}^{p} F_{i,j,k,h} F_{i,q,n,h} = \sum_{h=1}^{p} f_{i,j,k,h} f_{i,q,n,h} (1 - p_{i+1,h}^{p})^2 \tag{33}$$

If $j \neq q$, we have

$$c_{i,j,q,k,m} = \sum_{h=1}^{p} F_{i,j,k,h} F_{i,q,m,h} = \sum_{h=1}^{p} f_{i,j,k,h} f_{i,q,m,h} (1 - p_{i+1,h}^{p})^2 \tag{34}$$

Then, the blocking delay caused by packet competitions and flow controls can be denoted by (35):

$$\begin{aligned} & E(T)\Big(\sum_{q=1,q\neq j}^{p} \sum_{m=1}^{p} c_{i,j,q,k,m} N_{q,m} + \sum_{n=1,n\neq k}^{p} c_{i,j,j,k,n} N_{j,n} \Big) \\ & = E(T) \sum_{q=1,q\neq j}^{p} \sum_{m=1}^{p} \sum_{h=1}^{p} f_{i,j,k,h} f_{i,q,m,h} (1 - p_{i+1,h}^{p})^2 N_{q,m} \\ & \quad + E(T) \sum_{n=1,n\neq k}^{p} \sum_{h=1}^{p} f_{i,j,k,h} f_{i,j,n,h} (1 - p_{i+1,h}^{p})^2 N_{j,n} \end{aligned} \tag{35}$$

And the average waiting time is,

$$\begin{aligned} \tau_{i,j,k} = & E(T) \sum_{q=1,q\neq j}^{p} \sum_{m=1}^{p} \sum_{h=1}^{p} f_{i,j,k,h} f_{i,q,m,h} (1 - p_{i+1,h}^{p})^2 N_{q,m} \\ & + E(T) \sum_{n=1,n\neq k}^{p} \sum_{h=1}^{p} f_{i,j,k,h} f_{i,j,n,h} (1 - p_{i+1,h}^{p})^2 N_{j,n} \\ & + E(T) N_{j,h} + \frac{1}{2} \lambda_{i,j,k} E(T^2) \end{aligned} \tag{36}$$

VI. Experiment Results

A. Test method

A well-known, third-party NoC simulator, BookSim, is used to validate the accuracy of the proposed single-channel model. Originally developed for and introduced by the *Principles and Practices of Interconnection Networks* book [15]. BookSim's functionality has been continuously extended. We use BookSim 2.0, which supports a wide range of topologies, provides diverse routing algorithms and includes numerous options for customizing the router's micro-architecture.

BookSim cannot provide the flow-control feedback probability as a result directly; therefore we employ the following method:

At each simulation cycle, we record the flit number of any input queue; when an input queue is full, the flow control signal is regarded as issued till there is some free space. Then, the probability can be computed.

In this experiment, we adopt the XY deterministic routing and a 5x5 2D-mesh network. The observed results are obtained by simulating 2×10^7 cycles after a warm-up phase of 2×10^7 cycles, and then compared with analysis results. The performance of our model is analyzed under uniform traffic patterns, where any node transfers packets towards the destinations with equal probability.

The injection rate is specified in packets per cycle. For example, the injection rate is 0.25, which means each node injects a new packet every four cycles.

The error between the analytical results V_{test} and simulation results V_{sim} is calculated by the following equation.

$$Err = \frac{|V_{sim} - V_{test}|}{V_{sim}} \times 100\%$$

B. Accuracy validation for the computing method of flow-control feedback probability

In this section, we focus on the influence of input buffer size (B), number of pipeline stages (H_s) and packet size (P) on the flow-control feedback probability.

Simulation results in Fig.6 reveal that for the different design parameters, the trend of flow-control feedback probability is roughly similar: increasing slowly, increasing rapidly and tends to balance with the increased injection rates.

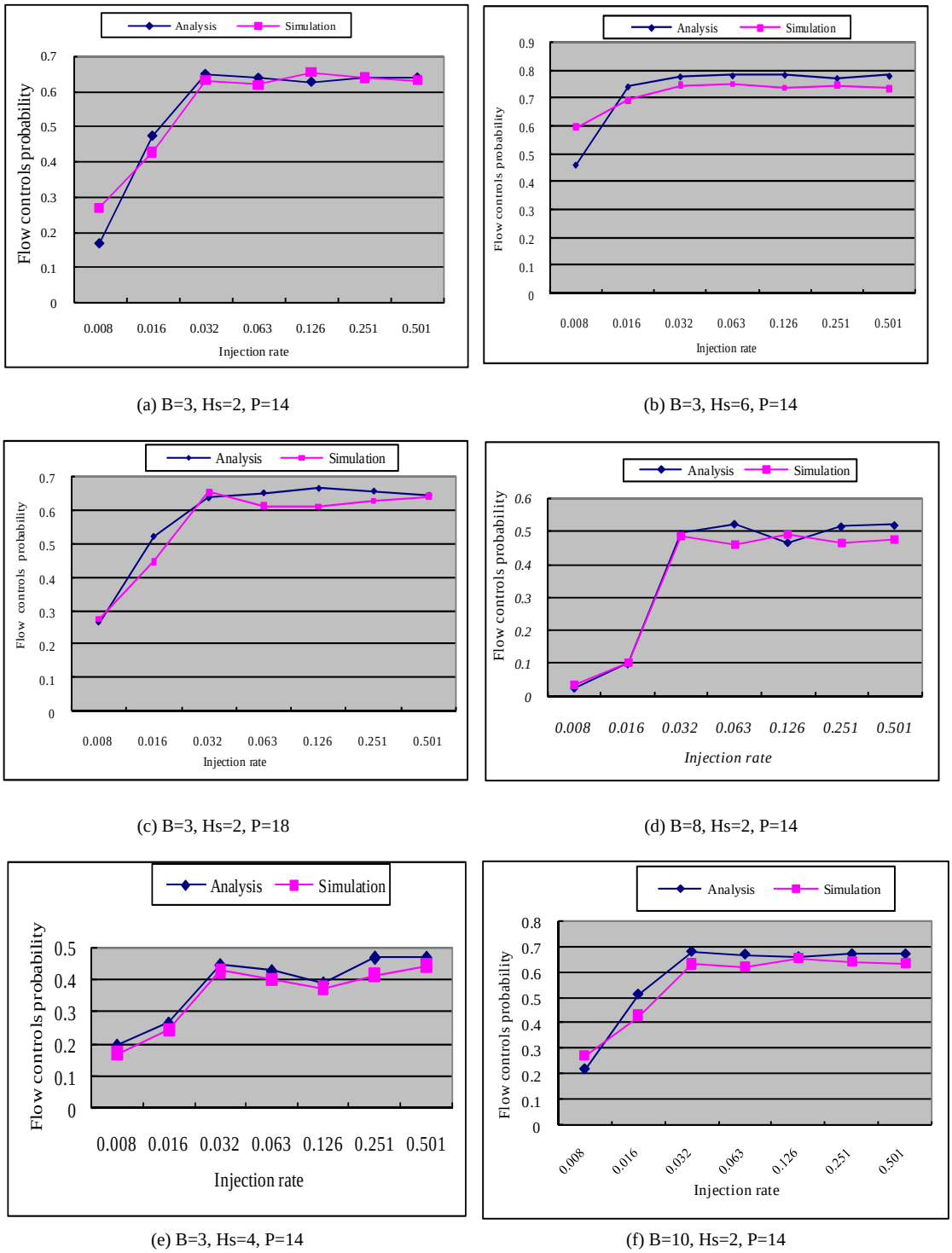
Our proposed analysis model tallies closely with the simulation results. When the injection rate increases to a certain extent, the traffic rates are much higher and buffers become full frequently. Therefore, the flow-control feedback probability increases significantly. If the rate increases continually, the network will be gradually saturated and lead to the equilibrium of flow-control feedback probability.

In detail, as Fig.6(a) shows, when the injection rate lies between 0.008 and 0.032, the probability changes sharply; as the injection rate continues to increase, the probability remains steady. The analysis results track the simulated closely and the mean error is 8.65%.

Comparing Fig.6(b) with 6(a), we can see that, when the rate is 0.016, the probability reaches steady. The mean error is 7.83%.

With the different packet size, the probability of flow-control feedback slightly changes, but the saturation point and the overall trend are similar, as shown in Fig.6(a) and 3(c). In Fig.6(c), the mean error is 6.22%.

As Fig.6(d) shows, when the input buffer size is 8, the probability is smaller than the other situations with low injection rates. It increases sharply to be steady as the injection rate lies between 0.016 and 0.032. We conclude that input buffer size is one of the most important factors which impact the probability as expected. The mean error is of 8.78%.



(a) B=3, Hs=2, P=14

(b) B=3, Hs=6, P=14

(c) B=3, Hs=2, P=18

(d) B=8, Hs=2, P=14

(e) B=3, Hs=4, P=14

(f) B=10, Hs=2, P=14

Figure 6. Flow-control feedback probability with different parameters

In summary, the computing method achieves a mean error of 7.87%, compared with the cycle-based simulation.

VII. Conclusion

A router model for NoC performance analysis is presented, which uses the M/D/1/B queuing theory to analyze various packet blocking-conditions. In addition, considering the effect of flow control to increase the blocking possibilities, a computing method of the flow control probability is proposed.

Experimental results show that the average error of the computing method for flow-control feedback probability is 7.87%.

In addition, we give the extended models for routers with multi-channel structure. Future work will refine the models and give their validation.

Acknowledgment

This research is supported by Chinese National Science Foundation under Grant 60773147, and Chinese National 863 High Technology Programs under Grant 2008AA01A204.

REFERENCES

[1] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection network", DAC, 2001, pp. 684– 689.
[2] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey and A. Sangiovanni-Vincentelli, "Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design", DAC 2001, pp. 667 –672.
[3] J. Duato, S. Yalamanchili, L. Ni, "Interconnection Networks: An Engineering Approach", Morgan Kaufmann, 2002.
[4] J. Kim, C.R. Das, "Hypercube communication delay with wormhole routing", IEEE Transactions on Computers C-43 (7) (1994) 806–814.
[5] O. Lysne, "Towards a generic analytical model of wormhole routing networks", Microprocessors and Microsystems 21(7-8), 1998, 491– 498.
[6] W. J. Dally, "Performance analysis of k-ary n-cube interconnection networks", IEEE Transactions on Computers, vol. 39, pp. 775-785, 1990.
[7] A. Khonsari, M. Ould-Khaoua and J. Ferguson, "A General Analytical Model of Adaptive Wormhole Routing in k-Ary n-Cube Interconnection Networks", SIMULATION SERIES, vol. 35, pp. 547-554, 2003.
[8] J. Hu, U.Y. Ogras and R. Marculescu, "System-level buffer allocation for application-specific Networks-on-Chip router design", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25(12), 2006, 2919-2933.
[9] Y.U. Ogras and R. Marculescu, "Analytical router modeling for networks-on-chip performance analysis", In: Proceedings of Design, Automation and Test in Europe Conference (DATE'07), Acropolis, 2007, 1096-1101.
[10] S. Foroutan, Y. Thonnart, R. Hersemeule and A. Jerraya, "An analytic method for evaluating network-on-chip performance", In: Proceedings of Design, Automation and Test in Europe Conference (DATE'10), Dresden, 2010, 1629-1632.
[11] H. Sarbazi-Azad, M. Ould-Khaoua, L.M. Mackenzie, "An accurate analytical model of adaptive wormhole routing in k-ary n-cube interconnection networks", Performance Evaluation 43 (2,3) (2001) 165–179.
[12] H. Sarbazi-Azad, M. Ould-Khaoua, L.M. Mackenzie, "On the performance of adaptive wormhole routing in the bi-directional torus network: A hotspot analysis", Microprocessors and Microsystems 25 (6) (2001) 277–285.
[13] H. Sarbazi-Azad, M. Ould-Khaoua, L.M. Mackenzie, "Analytical modeling of wormhole-routed k-ary n-cubes in the presence of matrix-transpose traffic", Journal of Parallel and Distributed Computing 63 (2003) 396–409.
[14] http://nocs.stanford.edu/cgibin/trac.cgi/wiki/Resources/BookSim.
[15] W.J. Dally and B. Towles, "Principles and Practices of Interconnection Networks", Morgan Kaufmann, San Francisco, 2004.
[16] Ming-che Lai, Lei Gao, Nong Xiao, Zhiying Wang. "An accurate and efficient performance analysis approach based on queuing model for network on chip". Proceedings of International Conference on Computer Aided Design, pp. 563-570, 2009.

Youhui Zhang received the BSc and PhD degrees in computer science from Tsinghua University, China, in 1998 and 2002 respectively. He is currently an associate professor in the Department of Computer Science at Tsinghua University. His research interests include computer architecture, storage systems and high-performance computing. He is a member of the IEEE and the IEEE Computer Society.

Xiaoguo Dong is a graduate student in the Department of Mathematical Science and Computing Technology, Central South University, China. His research interests include mathematical modeling and analysis.

Siqing Gan is a professor and Chair in the Department of Mathematical Science and Computing Technology, Central South University, China. His research interests include numerical solution of differential equations, and scientific and engineering computing.

Weimin Zheng received the BSc and MSc degrees in computer science from Tsinghua University, China, in 1970 and 1982 respectively. Now he is a Professor in the Department of Computer Science at the University of Tsinghua, China. His research interests include high performance computing, network storage and parallel compiler. He is a member of the IEEE and the IEEE Computer Society.

Cocktail method for BitTorrent traffic identification in real time

Zhe Yang
School of Computer Science and Technology, Soochow University, Suzhou, P.R.C.
Email: yangzhe@suda.edu.cn

Lingzhi Li, Qijin Ji, Yanqin Zhu
School of Computer Science and Technology, Soochow University, Suzhou, P.R.C.
Email: {lilingzhi, ji, yqzhu }@ suda.edu.cn

***Abstract*—Peer-to-peer (P2P) applications generate a large volume of traffic and seriously affect quality of normal network services. Accurate identification of P2P traffic is especially important for network management. The simplest method is based on port mapping. But dynamic port technique makes it ineffective. Signature-based approach is useless when facing encrypted traffic. Recently, some approaches use more complex machine learning and data mining algorithms relying on flow statistics or host behaviors. Due to the sophisticated algorithms, they need a time-consuming process for training or calculating, they can hardly be used in real-time identification. In this paper, we propose a cocktail approach consists of three sub-methods to identify BitTorrent (BT) traffic. We apply application signatures to identify unencrypted traffic. And for those encrypted flows, we propose the message-based method according to the features of the message stream encryption (MSE) protocol. At last, we propose a pre-identification method based on signaling analysis. It can predict BT flows and distinguish them even at the first packet with SYN flag only. And we use modified Vuze clients to label BT traffic in real traffic traces, which help us to make high accuracy benchmark datasets to evaluate our approach. The results illustrate the effectiveness of our approach, especially for those un- or semi- established flows, which have no obvious signatures or flow statistics.**

***Index Terms*—Peer-to-peer, traffic identification, application signature, message stream, signaling analysis, benchmark dataset**

I. INTRODUCTION

In recent years, P2P protocol is widely used in many network applications, which generate 50-70% of the Internet traffic [1]. They greatly consume network bandwidth and seriously affect QoS of normal network activities. Thus, accurate identification of P2P traffic is especially important for network management and traffic optimization. But, P2P traffic is more difficult to be identified because of its quite different communication model. Furthermore, some newer-generation P2P applications apply various strategies to avoid detection, such as masquerade, obfuscation and encryption.

To identify the traffic of P2P applications, the simplest method is port-mapping on transport-layer. However, it becomes less accurate and ineffective, since P2P applications do not rely on those well-known ports anymore, and usually use dynamic port numbers. Another method is payload-based analysis, which searches packet payload for the application signatures or the sharing data blocks by TCP/UDP flows. But, this method is ineffective in identifying encrypted P2P traffic. Recently, some researchers try to use machine learning algorithms to identify P2P traffic with flow statistical information. But different flows, such as TCP and UDP, may have different statistical characteristics. And the statistical characteristics may be unstable because of the dynamic network situation. Instead of analyzing individual flows, a new method is based on the patterns of host behavior at the transport layer. It pays attention to all the flows generated by a specific host. But this method must gather information from several flows for each host before it can decide whether the host runs P2P applications. And like per-flow based method, it also needs a training process, and in many cases due to its sophisticated algorithms, it can hardly be used for real-time identification. Furthermore, for any identification method, benchmark traces containing prior labeled traffic of the P2P application are necessary to verify the effectiveness. But it is difficult to get 100% accuracy.

To resolve these problems, we propose a cocktail approach in our previous work [35], to identify BitTorrent (BT) traffic, the dominating one of P2P file-sharing applications, which has three sub-methods. This paper gives more explanations on identifying different kinds of BT traffic at the very beginning of or even before the BT flows generated.

a) Sub-method one: signature-based identification. It is used to identify the unencrypted BT traffic. Though BT applications have applied the message stream encryption (MSE) protocol to obfuscate the traffic, there are still over one third of them unencrypted [2]. So we recall this veteran, and call it *M1*.

Manuscript received December 27, 2010; revised March 24, 2011; accepted May 31, 2011.
Supported by the NSFC under Grant No. 61070170, the NSFC of Jiangsu under Grant No.BK2009589.
Corresponding author: Zhe Yang (yangzhe@suda.edu.cn).

doi:10.4304/jcp.7.1.85-95

b) Sub-method two: message-based identification. For those encrypted BT traffic, we resemble the bidirectional TCP flows into message streams. And if the direction and length of first three messages satisfy the criteria of MSE, this flow is identified as encrypted BT traffic. In our work, it is called *M2*.

c) Sub-method three: signaling-based pre-identification. Theoretically, *M1* and *M2* can identify the almost all BT traffic, un- and encrypted. But, matching the signatures or resembling the bidirectional flows are time-consuming tasks. And they may still miss the first few packets of the flows. Thus, we propose the pre-identification method based on signaling packets analysis, called *M3*. It can can predict BT flows and identify them at the first packet with *SYN* flag only.

Besides the approach itself, we also propose an easy method to make benchmark traces with almost 100% accuracy. Some modified Vuze clients are used to report how much traffic on earth is generated themselves.

In summary, the main contributions of this paper are as follows.

First, we propose a cocktail identification approach consists of three sub-methods. They combine with each other and aim at different BT traffic identification. Especially, the signaling packets are analyzed, which help us to predict the upcoming data flows. Results indicate that near 84% flows can be identified at the first packet and in 1ms, even they have not finished the 3-way TCP handshake. Actually, some of them are un- or semi-established TCP connections, which always are omitted by other approaches because they have no obvious signatures or flow statistics. Thus this cocktail approach achieves the ability of real-time identification with high accuracy and low overhead.

Second, we use modified Vuze clients to evaluate our approach, not only to generate real BT traffic, but also to label out these traffic in benchmark traces by themselves. It can reach near 100% accuracy and help us to well evaluate our cocktail approach.

The rest of the paper is organized as follows. We review the related works in Section 2. Section 3 explains our cocktail approach in details. In Section 4, we explain how to establish our experimental environment and define some metrics. And the evaluation results are given in Section 5. Finally, we discuss and conclude our work in Section 6.

II. Related Works

A. Port-based Identification

Earlier P2P applications, like many Internet services such as Web, Email and FTP, mostly run on the default ports, e.g. TCP 6881~6889 for BT, TCP 6346~6347 for Gnutella. So the simplest method for detection systems and firewalls to identify, control or even block the corresponding P2P traffic is based on these pre-defined ports [3-5]. But to circumvent the firewalls and detection systems, P2P applications intentionally disguise their traffic by using other applications' ports, even port 80, or random ports at each communication. Thus, this traditional port-based technique becomes less accurate and ineffective.

B. Payload-based Identification

Payload-based method searches the packet payload at application layer to find some common features.

On kind of feature are application signatures, which are the common strings in P2P protocols. These signatures often appear in the signaling and handshake packets before the data transmission. And they are useful in real time identification of network traffic [6-8]. This signature-based method achieves high accuracy once and is employed in some commercial network management products. However it also has some limitations. First, it is ineffective when P2P traffic is encrypted. Second, a lot of P2P applications use proprietary protocols. Lacking open protocol specifications makes finding appropriate signatures difficult. And maintaining up-to-date signatures for various newly emerged P2P applications are daunting tasks.

In addition to signature-based method, some other payload-based methods are proposed, which inspect the data transfer behavior in P2P applications. One basic observation is that a P2P peer uploads data to others after downloading it. Lu et al. [9] store the first *K* bytes of each packet in downloading flows for each host. When the same content is found in uploading flows of the host, these flows are identified as P2P flows. ACAS [10] uses the first *N* bytes of payload as input to train a machine learning model to classify flows. But only searching the some first bytes of payload to find the shared data is poor to identify some P2P applications. In [11], Xu et al. search shared data in the whole payload to solve the problem. They divide payloads of flows into data blocks. For flows sharing the same data blocks are identified as P2P traffic. Levchenko et al. [12] build several probabilistic models on payload, including the statistical model treating each n-byte flow distribution as a product of *N* independent byte distributions and the Markov process model which relies on introducing independence between bytes. These approaches are quite impractical in real-time identification systems. Because they will not identify any flows until the stored data in downloading flows reappears in uploading flows later. Storing and finding the shared data is memory- and time-consuming.

C. Flow-based Identification

Recently, some methods use machine learning algorithms to classify network traffic by using flow statistical information [13-23]. They develop discriminatory criteria based on statistical observations of various flow properties, such as the packet size distribution per flow, flow duration, mean inter-arrival times between packets, and number of upload/download packets/bytes, etc. And then they employ classification, clustering and other machine learning algorithms to assign flows to classes. The algorithms can be further summarized into supervised and unsupervised methods.

Moore et al. [14] apply Bayesian analysis techniques to categorize traffic by application. Roughan et al. [15] classify traffic flows into four classes suitable for quality

of service. They demonstrate the performance of nearest neighbor and linear discriminant analysis algorithms. Zander et al. [16] compare several supervised algorithms, including Naive Bayes, C4.5 decision tree, Bayesian Network and Naive Bayes Tree. They find that the classification precision of the algorithms is similar, but computational performance can be significantly different. In addition to these supervised classification techniques, unsupervised clustering methods are also used [17-20]. Campos et al. [21] apply some different hierarchical clustering methods to identify groups of similar communication patterns. Similarly, McGregor et al. [22] use the expectation maximum algorithm to cluster the flows. The experiment finds that the average precision of classification is very high, but some applications are difficult to distinguish.

Most of these methods only focus on the statistics of a single flow. And there are four challenges in identifying P2P traffic by using per-flow properties. First, in order to generate the appropriate traffic classes, these machine learning algorithms need a training process with labeled sample traces. Since network situations are quite different and dynamic, flow statistical characteristics in different sample traces are quite different. So the traffic classes relying on different sample traces may be not suitable for any networks in any time. Second, different flows in same application may have different statistics and even use different transport protocols. For example, in BT applications, some flows are used to get peer's information, some for negotiation between peers, and others for data transferring. Third, some flows may have no obvious or specific statistics. If only looking at per-flow statistics, flows belonging to different applications may have similar characteristics. Fourth, in many cases due to their sophisticated algorithms, they can hardly be used in real-time identification.

D. Behaviour-based Identification

Another identification approach is based on the patterns of host behavior or special communication patterns in transport layer. Instead of analyzing individual flows, it pays attention to all the flows generated by a specific host. Karagiannis et al. [23] attempt to capture the inherent behavior of a host at three levels: the social, functional and application levels. This approach mainly focuses on higher level communication patterns such as the number of source ports a particular host uses for communication. Constantinou et al. [24] investigated some fundamental characteristics of P2P applications, such as the huge network diameter and the presence of many hosts acting as both servers and clients, to identify P2P traffic. Xu et al. [25] use information theoretic and data mining techniques to build behavior profiles of Internet backbone traffic. Hu et al. [26] propose a method to build behavioral profiles of the target application which then describes the dominant communication patterns of the application.

However, this behavior-based method can not identify a single flow and is time-consuming, since it must gather information from several flows for each host before it can decide the role of a host. And it is hard to find a general pattern for all P2P applications. Like flow-based approaches, this method can hardly be used by real-time systems either.

E. Real-time Identification

In order to identify traffic in real time, some researchers try to only use the first *N* packets or first *M* bytes of each flow. Haffner et al. [10] automate the construction of protocol signatures by employing three supervised machine learning approaches on traffic containing known instances of each protocol, and relying solely on first 64 bytes of each flow. Bernaille et al [27] use the first 5 packets in each TCP flow to classifying traffic [18]. Li et al. apply decision tree algorithms using the first 5 packets of flows to identify the P2P traffic in earlier stage of the TCP connections. Ma et al. [28] develop three alternative mechanisms with statistical and structural content models for automatically classifying traffic into distinct protocols based on correlations between the packets with only first 64 bytes of each flow.

However these methods also have some limitations. The first *N* packets and first *M* bytes of each flow may have significant differences for the same P2P application, since some peers may re-send some packets because of the unreliable network situation. And the packets in the same position of the corresponding flows may also be different. Therefore in this case the positions of packets and bytes are not reliable information sources to identify a flow.

III. COCKTAIL IDENTIFICATION OF BT TRAFFIC

In this work, we combine three sub-methods as a cocktail approach to identify the traffic of BT applications, including signature-based, message-based and signaling-based pre-identification, donated as *M1*, *M2* and *M3* separately. In this section, we explain these three sub-methods.

A. M1: Signatures-based Identification

We first use traditional signature-based method to identify the unencrypted BT traffic. Though BT applications use the MSE protocol to obfuscate the traffic, there are still some parts of them unencrypted. According the comparative numbers in Germany and Southern Europe, the relative amount of unencrypted BT traffic in 2008 were still over 77% and 74% [1]. And in our pervious work [2], there are still have one-third of all BT clients establish unencrypted connections to others. So in this paper, we still recall this veteran, and call it *M1*.

The BT protocol signatures are identified from previous studies [7, 23], public BT protocol specifications [29], and by reverse engineering. The signatures for different BT protocols are represented by regular expressions, as shown in Table 1. Besides the common used signature "^\x13BitTorrent protocol" for identifying the data transferring protocol (peer wire), we pay more attention to another three protocols. Although they will not generate as much traffic as peer wire protocol, they play the critical roles in BT applications. They are necessary to maintain the BT system and exchange the

TABLE I.
APPLICATION SIGNATURES OF BITTORRENT PROTOCOLS

Protocol Name		Signatures	Protocol
Peer Wire		^\x13BitTorrent protocol	TCP
TCP Tracker		^get .*announce?info_hash	TCP
UDP Tracker		^\x00 00 04 17 27 10 19 80	UDP
DHT	request	^d1:a	UDP
	response	^d1:r	
	error	^d1:e	

peer information for data transferring, we call them signaling protocols. The purpose that we use these signaling protocols' signatures is not only for identifying their traffic, but also for obtaining the peers' information which is used by *M3*.

B. M2: Message-based Identification

Second, we use message-based method to identify the encrypted BT traffic, we call it *M2*. It is similar to the flow-based and behavior-based methods, but it still has some differences. Before we explain the *M2* in detail, we first discuss the MSE protocol.

1) MSE protocol analysis

MSE [30] is designed to provide security features to the BT at a lower layer by acting as a wrapper. It behaves randomly, both in terms of flow behavior (such as packet sizes) and in terms of application data. This makes it almost impossible to detect the MSE protocol with flow-based methods. Two communicating peers start the handshake of MSE protocol after a completed 3-way TCP handshake. The handshake of MSE is separated into 5 blocking steps, as shown in Fig. 1.

To achieve complete randomness from the first byte, initiating peer (denoted as A) and responding peer (denoted as B) use a D-H key exchange [31], which uses large safe prime P, to create a session key in step 1 and 2. After negotiating the D-H key, the rest of the handshake is encrypted using RC4 [32], with the first 1024 keystream bytes discarded. The RC4 key is based on both the D-H key and a weak shared secret which is the "InfoHash" value transmitted in the .torrent file. The payload stream is either encrypted using RC4 or sent in plaintext. The desired payload delivery method is negotiated in step 3 and 4. In step 3, A sends the method it is willing to use in *CryptoProvide* and B selects the desired method in *CryptoSelect*. Although MSE protocol makes it no chances to be detected, there are some distinctive properties that can be observed in the handshake process, especially in the first three steps.

$$1.\ A \to B\ :\ g^a \bmod P, PadA$$
$$2.\ B \to A\ :\ g^b \bmod P, PadB$$
$$3.\ A \to B\ :\ H('req1',S), H('req2',Skey) \oplus H('req3',S),$$
$$E(VC, CryptoProvide, len(PadC), PadC, len(IA)), E(IA)$$
$$4.\ B \to A\ :\ E(VC, CryptoSelect, len(PadD), PadD), \hat{E}(Payload\ Stream)$$
$$5.\ A \to B\ :\ \hat{E}(Payload\ Stream)$$

Figure 1. Handshake of MSE

In first step, peer A starts the MSE handshake by sending a D-H public key Y_a ($Y_a = g^a \bmod P$) and some padding (*PadA*) to peer B. In MSE, g is a generator and it is used to generate Z_P^* with safe prime P and its value is 2. a is a D-H exponent, which is a variable size random integer and MSE recommend 160 random bits. P is a 768 bit safe prime. Thus, the size of Y_a is 96 bytes (768/8=96 bytes) and its value is not bigger than P. And *PadA* is random data with a random length of 0-512 bytes. So, the length of message sent from A to B in step 1 will always be in the range of 96 to 608 bytes.

Next, in step two, peer B responds with its D-H public key Y_b ($Y_b = g^b \bmod P$) and some padding (*PadB*). Similarly, the length of message reply from B to A in step 2 is also in the range of 96 to 608 bytes.

In the message sent by peer A in step 3, *H* is the SHA-1 hash function with 20 bytes binary outputs. *S* is the negotiated D-H key. *Skey* is the weak shared secret key mentioned before. *E*() is RC4 encryption, with key *H*('*keyA*', *S*, *SKey*) for A and *H*('*keyB*', *S*, *SKey*) for B. *VC* is a verification constant, but in this version of BT protocol it is a string of 8 bytes set to 0x00. *CryptoProvide* and *CryptoSelect* are a 4-byte bitfields. *len*(*X*) specifies the length of *X* in 2 bytes. *PadC* and *PadD* are zero-valued padding of 0–512 bytes. *IA* is the initial payload data from A and it is considered as atomic. Thus there must be no blocking operations before *IA* is completely transmitted. But after that, a block occurs since A waits for B to send next message before A continues to send his bitfield, thus *IA* only includes the following fields (in order): PStrLen (1 byte), PStr (19 bytes), Reserved (8 bytes), InfoHash (20 bytes) and PeerID (20 bytes). Thus the length of *IA* is 68 bytes. So, the total length of the third message is in the range of 124 to 636 bytes.

2) MSE message resembling

Although, the first three messages of MSE handshake have some distinctive properties in length. But unfortunately, each of them is not transmitted in one packet. It is often divided into several packets and the payload size is around 100 bytes per packet. Thus we

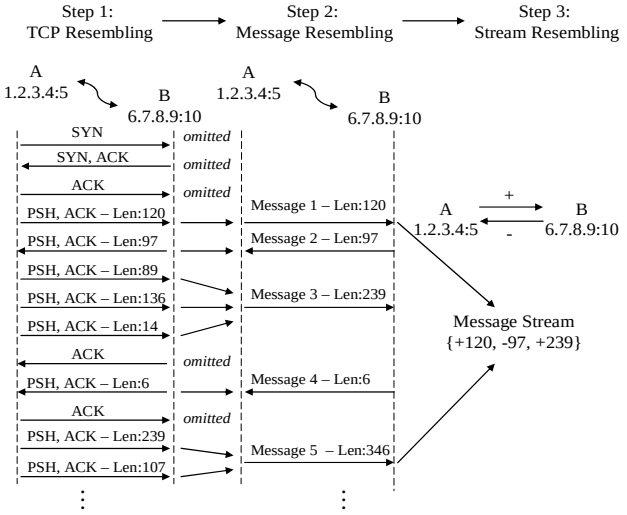


Figure 2. MSE message resembling

need to resemble packets in a TCP flow into messages first, and then determine whether this flow belongs to encrypted BT traffic according to direction and length of first three messages. Fig. 2 describes the main steps.

Step 1: TCP Resembling. First we group the traffic packets into TCP flows with the four tuples {*SrcIP*, *SrcPort*, *DestIP*, *DestPort*}. The source and destination of the flow is defined as the source and destination IP address of the flow's first packet.

Step 2: Message Resembling. Because MSE handshake messages are often divided into several packets, so we merge first few packets into messages according to the direction and the flags. In each flow, we only take the packets with the *PSH* flag into consideration. While those *ACK*, *RST* and *SYN* packets are omitted. We merge the *PSH* packets with same direction into one message, until meet the next reverse *PSH* packet. And the message length is the sum of payload length of these packets.

Step 3: Stream Resembling. Finally, we arrange these messages into a stream. A message stream is defined as a bidirectional, ordered sequence of messages. The source and destination of a stream is same as the flow. In each stream, if a message is sent from the source to the destination, then the message direction is positive and denoted as "+". Similarly, the message direction is negative if the message is sent from the destination to the source and denoted as "-". And because only the first three messages in MSE handshake have distinctive properties in length, so the message stream only include the first three messages in each flow.

Now the flow form A (1.2.3.4:5) to B (6.7.8.9:10) can be described in a message stream {+120, -97, +239}. Then we compare it with the stream of a normal MSE handshake process. According to aforementioned analysis, the stream of a normal MSE handshake should be {+[96,608], -[96,608], +[124,636]}. Thus, this flow between A and B can be identified as an encrypted BT traffic.

C. M3: Signaling-based Pre-identification

For a distributed application system, like BT, the most important thing to keep the system alive and running well is that all users may discover each other easily. So, the system must have a mechanism to maintain the information of all users. In BT, a peer can obtain the list of other pees from trackers, DHT and by peer exchange (PEX) protocol [34]. Thus, if any identification method can grasp the list of peers, it may find the Achilles heel.

1) Peer list from tracker

Tracker is very critical for BT. After downloading the .torrent file, peers contact the specified tracker to request the current list of peers. The request and response

```
0000  00 22 64 6b fb f9 00 1e  08 08 0c 60 08 00 45 00   ."dk.... ...`..E.
0010  02 d3 b2 f5 40 00 2e 06  80 9a da ca e3 1b c0 a8   ....@... ........
0020  98 06 1f 95 04 83 6a d0  44 1c 92 9d 31 87 50 19   ......j. D...1.P.
0030  19 20 82 2f 00 00 48 54  54 50 2f 31 2e 30 20 32   . ./..HT TP/1.0 2
0040  30 30 20 4f 4b 0d 0a 0d  0a 64 31 30 3a 64 6f 6e   00 OK... .d10:don
0050  65 20 70 65 65 72 73 69  35 32 34 65 38 3a 69 6e   e peersi 524e8:in
0060  74 65 72 76 61 6c 69 31  38 30 30 65 39 3a 6e 75   tervali1 800e9:nu
0070  6d 20 70 65 65 72 73 69  32 31 35 36 65 35 3a 70   m peersi 2156e5:p
0080  65 65 72 73 36 30 30 3a  3d b2 15 9f 00 50 7d 27   eers600: =....P}'
0090  3f 02 00 50 75 41 2d d5  00 50 72 eb 54 2a 00 50   ?..PuA-. .Pr.T*.P
00a0  de 5e de b9 41 f1 de 47  82 8d 46 f4 4a 0c 81 70   .^..A..G ..F.J..p
00b0  40 3f de de 4d 66 00 50  45 56 7b d9 00 50 da 0d   @?..Mf.P EV{..P..
00c0  ae 9b 00 50 3b bc 99 9a  7f b3 76 75 0a 25 34 c9   ...P;... ..vu.%4.
```

Figure 3. Peer list from tracker over TCP

TABLE II.
STRUCTURE OF ANNOUNCE RESPONSE PACKET

Offset	Size	Name	Value
0	32-bit integer	action	1(announce)
4	32-bit integer	transaction_id	
8	32-bit integer	interval	
12	32-bit integer	leechers	
16	32-bit integer	seeders	
20+6*n	32-bit integer	IP address	
24+6*n	16-bit integer	TCP port	

message between peers and trackers is transferred by TCP or UDP protocol.

The TCP based tracker responds peer's GET requests with "text/plain" document consisting of a bencoded dictionary which has the following parameters: failure reason, interval, complete, incomplete, peers, etc. The value of parameter of "5:peers" is a list of dictionaries containing a list of peers. To reduce the size of responses and to reduce memory and computational requirements, trackers use a compact format of peer list, where each peer is represented using 6 bytes. The first 4 bytes contain the 32-bit IPv4 address. The remaining two contain the port number. Both address and port use network-byte order. So if we identify a TCP flow as a TCP_Tracker flow, we mark it. And after some time, if we receive a reverse packet in same flow which containing the parameter of "5:peers", we will obtain the list of peers. For example, in the TCP packet showed in Fig. 3, we match the parameter "5:peers". And behind the ":", there is a 600 bytes list of peers. Each peer is represented using 6 bytes. So we can find the first two peers are "0x 3d b2 15 9f 00 50" (61.178.21.149:80) and "0x 7d 27 3f 02 00 50" (125.39.63.2:80).

In UDP based tracker protocol, there are several types of packets, including connect request, connect response, announce request, announce response, scrape request, scrape response, error, etc. The list of peers is only in the announce response packets from tracker to peers. The structure of the announce response packets is shown in Table 2. And we can find that the value of the first 4 bytes always equals 1, according to the protocol specification. It can be used as an additional signature to identify the announce response packets. But this signature is too ordinary, and may introduce lots of false positive. Thus we first use the signature "^\x00 00 04 17 27 10 19 80" to identify a UDP flow as an UDP_Tracker flow. And if we receive a reverse packet in the same UDP flow and its first 4 bytes are "0x 00 00 00 01", we extract the list of peers from the twentieth bytes to the end.

2) Peer list from DHT

Because of the importance of tracker in BT, so it is easy for ISP to limit the BT traffic by block the trackers. To implement the trackerless system, BT uses a DHT for storing peer's information. The DHT protocol is a simple RPC mechanism consisting of bencoded dictionaries sent

```
0000  00 1e 08 08 0c 60 00 16  47 88 52 bf 08 00 45 00   .....`.. G.R...E.
0010  00 9b 4c d2 00 00 6b 11  12 5a be e0 d8 94 c0 a8   ..L...k. .Z......
0020  98 08 58 bd 1a e1 00 87  56 ae 64 31 3a 72 64 32   ..X..... V.d1:rd2
0030  3a 69 64 32 30 3a aa 5a  23 ad 0c e8 9e 0f b1 df   :id20:.Z #.......
0040  4a 95 bb 24 88 f0 ae 83  08 de 35 3a 74 6f 6b 65   J..$.... ..5:toke
0050  6e 32 30 3a 82 ea b5 bb  69 67 aa 23 26 b5 7e 39   n20:.... ig.#&.~9
0060  62 7f 1b d1 fc 1f fc 32  36 3a 76 61 6c 75 65 73   b......2 6:values
0070  6c 36 3a da 04 bd 17 1a  e1 36 3a 44 70 f1 b2 26   l6:..... .6:Dp..&
0080  e8 36 3a 7c 08 4f b0 36  35 36 3a 75 54 61 b6 00   .6:|.O.6 56:uTa..
0090  50 36 3a 3c df ff 05 53  80 65 65 31 3a 74 32 3a   P6:<...S .ee1:t2:
00a0  05 68 31 3a 79 31 3a 72  65                       .h1:y1:r e
```

Figure 4. Peer list from DHT over UDP

over UDP. A single query packet is sent out and a single packet is sent in response. There are three message types: query, response, and error. And there are four types of queries: ping, find_node, get_peers, and announce_peer. The list of peers is only in the response packet to "get_peers" query. If a queried node knows some peers, it returns the compact format peer information in a parameter "6:values" as a list of peers. So we use the signature "d1:r" to identify a UDP packet as a DHT response message. And if it has the parameter of "6:values", we will get the list of peers. For example, in the UDP packet showed in Fig. 4, we match the signature "d1:r", so it is a DHT response message. And we also match the parameter "6:values". And behind it, there is a list containing 5 peers. Each peer is represented using 6 bytes. So we can find the first two peers are "0x da 04 bd 17 1a e1" (218.4.189.23:6881) and "0x 44 70 f1 b2 26 e8" (68.112.241.178:9960).

3) Peer list from PEX

Most BT clients also use PEX protocol [33] to gather peers information more swiftly and efficiently. PEX greatly reduces the reliance of peers on a tracker by allowing each peer to directly update others in the swarm. PEX messages are exchanges periodically over peer wire protocol, which are bencoded dictionaries containing a list of peers to be added and removed. The parameters of "added" and "added6" are used to add one or more IPv4/6 addresses. In this work, we only consider the "added" field that contains the IPv4 addresses. To avoid the confusion with "added6", we use the string "5:added" instead of "added" to locate the peer list. Thus we first use the signature "^\x13BitTorrent protocol" to identify a TCP flow as a Peer_Wire flow. And in the same TCP flow, if we receive any packets containing the string of "5:added" in application payload, we extract the peers information.

4) Pre-identification of BT traffic

After obtaining the list of peers from tracker, DHT and PEX, we store them into a hash table, named list of pre-identified flow. The hash value of two tuples {*IP*, *Port*} of each peer extracted from the peer list is calculated with MD5 algorithm. For any newly arrived TCP flow, if the hash value of {*DestIP*, *DestPort*} has already in the list of pre-identified flow. This flow should be marked as BT traffic. But with the time elapsing, we will get more and more list of peers, thus list of pre-identified flow will expand rapidly. So some records in the list should be eliminated periodically. We used the LRU (Least Recently Used) algorithm to update the list of peers, and the update interval is an empirical value and will be discussed in section IV.

IV. Experimental Setup

In this section, we explain how to establish our experimental environment, make the benchmark traces and setup the update interval of list of pre-identified flow.

A. Environment and Traces

Another important point in traffic identification is to verify the approaches with benchmark traces containing accurate labeled traffic. Basically there are two methods to obtain traffic traces. One is to capture real traffic from the Internet and then label the target application with third party tools. But it is very difficult to reach 100% accuracy. Another method is to get self-generated traffic traces of the target applications in a controlled environment. This method can get higher accuracy.

Thus, we collect 7 real traffic traces at the up-link line of our lab, a total controlled environment, including self-generated BT traffic with modified Vuze clients [2]. In each trace, we used 10 computers to run the modified Vuze clients with 5 different .torrent files each to generate the real BT traffic, 5 for unencrypted and 5 for encrypted. The download speed is limited to 1KB/s and the upload speed, 10KB/s. We use these modified Vuze clients not only to generate real BT traffic, but also to enable them to label the BT traffic they generated. And we use another 10 computers to generate background traffic, such as HTTP(S), FTP, DHCP, SMTP and POP3, etc. In all traces, there are little packets generated by Windows OS or system services. All these packet traces are collected at the uplink line of our lab. For privacy, we anonymized all the IP addresses. Those .torrent files running on modified Vuze clients come from the www.torrentz.com, and belong to five categories: movie, TV, music, game and anime. And in each trace, every client randomly selects 5 different .torrent files.

Then we divided these 7 traces into two types, 2 for configuration and 5 for evaluation. The two configuration traces were collected from March 1 to 5, 2010 (denoted as C1) and March 11 to 15 (C2). We first apply our cocktail approach on these configuration traces offline to determine the update interval of list of pre-identified flow. And the five evaluation traces are collected from April 1 to 9, 2010 (denoted as T1), April 11 to 19 (T2), April 21 to 29 (T3), May 1 to 9 (T4) and May 11 to 19 (T5). After collecting these real traffic traces, we label them as benchmark traces.

B. Benchmark Traces

Like other research works, well labeled benchmark traces are very critical to verify the effectiveness of our approach. Because there are few benchmark traces containing accurate labeled traffic of the target BT applications, so we use two methods to label the BT traffic in real traffic traces, including signature-based methods and manual analysis.

In the signature-based method, the signatures we used are listed in Table 1. In this step, we process traces with Bro [35], an open-source network intrusion detection system. We added functions to Bro to match the signatures in packet payloads and label the corresponding

TABLE III.
NUMBER OF FLOWS/PACKETS/BYTES IN THE BENCHMARK TRACES

Protocol		C1	C2	T1	T2	T3	T4	T5
BT	flows	472k	597k	604k	815k	773k	761k	702k
	packets	19,826k	21,837k	20,375k	18,223k	23,148k	12,492k	14,910k
	bytes	4,133M	2,555M	4,107M	5,497M	5,644M	3,371M	5,111M
non-BT	flows	176k	257k	342k	297k	425k	259k	393k
	packets	10,862k	8,093k	7,295k	9,745k	8,624k	7,065k	7,058k
	bytes	2,663M	1,129M	1,454M	2,992M	2,787M	2,242M	2,327M

flows. Thus, the flows of unencrypted data transferring, TCP-Tracker, UDP-Tracker and DHT can be labeled accurately.

But not all BT flows have traceable signatures, especially for those encrypted BT traffic. And because there still have no well accepted methods, which can identify the encrypted BT flows with 100% accuracy, we do it manually. Actually, the best way to label the traces with high accuracy is to let the BT applications tell us how much traffic in the traces on earth is generated by themselves. So we label the traces with the help of those modified Vuze clients we used to generate the real BT traffic. We choose Vuze, formerly Azureus, because it is the most popular open source client with cross-platform compatibility and features a powerful API for dynamically adding new functionality. Like other ordinary BT clients, these modified Vuze clients can parse the .torrent files, query and connect to other peers, and download the sharing files. Moreover, we enable them to report some information of all other peers, which they try to connect to or receive the connections from.

But they only report the IP address and port of peers, so we still can not labeled BT flows directly. Thus we need further processing at these traffic traces. First we resemble the packets into flows according to the five tuples {*SrcIP*, *SrcPort*, *DestIP*, *DestPort*, *Protocol*}. For each flow, if the {*SrcIP*, *SrcPort*} or {*DestIP*, *DestPort*} is reported by modified Vuze clients, thus we label this flow as BT traffic. For example, if a modified Vuze client reports that it connected to a peer at {1.2.3.4, 5}. And in real traces, there are two flows {6.7.8.9, 10, 1.2.3.4, 5, TCP} and {10.9.8.7, 6, 1.2.3.4, 5, TCP}. Thus, we label these two flows as BT traffic.

And for those background traffics, such as HTTP(S), FTP, DHCP, SMTP, POP3, and those generated by Windows OS, system services and even known applications, we all labeled them as non-BT traffic. Finally, we labeled the two configuration traces, C1 and C2, and five evaluation traces, T1 to T5, as our benchmark traces. And the summary of these traces is given in Table III.

C. Metrics Definitions

We use three kinds of metrics to evaluate the accuracy of our method at the level of flow, packet and byte. First, we give the metrics of flow with examples. For example, if there are *n* flows as total in a traffic trace. So if a flow is identified as the BT flow but in fact it does not belong to it, we call it a false positive and use n_{fp} denotes the total number of false positive flows. Similarly if a BT flow is not identified, we call it a false negative and use n_{fn} denotes the total number of false negative BT flows.

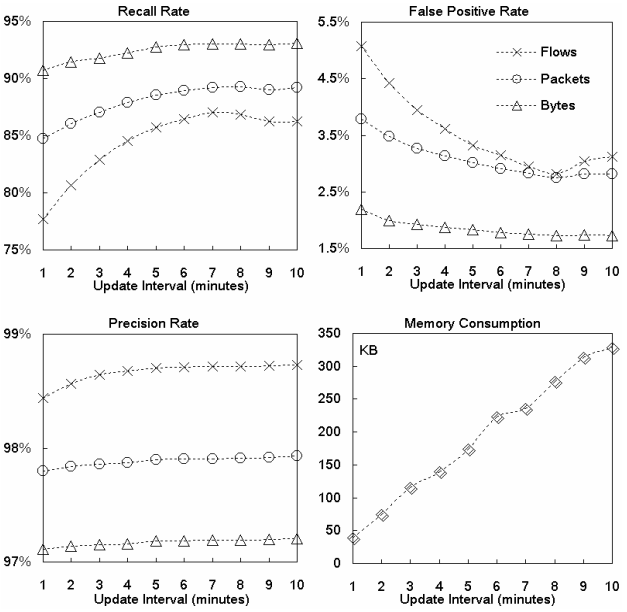


Figure 5. Update interval setup

And if it is correctly identified, we call it a true positive and use n_{tp} denotes the total number of true positive BT flows.

So the fist metric we used is the *recall rate of flows* (R_f). It is used to measure the portion of the flows that can be identified correctly by our method out of the all flows belonging to the BT application.

$$R_f = n_{tp} \Big/ \left(n_{fn} + n_{tp} \right) \tag{1}$$

The second one is the *precision rate of flows* (P_f). It is the number of the true positive flows as the fraction of the total flows we identified

$$P_f = n_{tp} \Big/ \left(n_{tp} + n_{fp} \right) \tag{2}$$

The third one if the *false positive rate of flows* (FP_f). It is used to measure the portion of non-BT flows but identified as BT flows out of the all non-BT flows.

$$FP_f = n_{fp} \Big/ \left(n - n_{fn} - n_{tp} \right) \tag{3}$$

Similarly, we use R_p, P_p and FP_p denote the three metrics of packet, R_b, P_b and FP_b for byte.

D. Update Interval Setup

We first applied our approach on configuration traces offline. And after considering the recall, precision, false positive and the memory consumption of list of pre-identified flow, we determined the update interval of list of pre-identified flow. Fig 5 shows the average value of four metrics on two configuration traces, C1 and C2. And we can find that when the update interval is 7 minutes, the *recall rates* of flows, packets and bytes can reach the maximum at the same time. While the *false positive rates* of flows, packets and bytes all reach the minimum when the update interval is set as 8 minutes. The *precision rates* are not influenced obviously by the parameter of update interval. But when the interval increases, the memory consumption of list of pre-identified flow increases simultaneously. It is almost a linear increase. A good identification approach should have low *false positive*

rate, high *precision rate* and *recall rate*. So we set the update interval at 8 minutes in following evaluation experiments.

V. Evaluation Results

In this section, we present the results of identifying BT traffic in the evaluation traces, T1~T5. Among our three sub-methods, *M1* and *M2* are enough to identify all BT traffic theoretically. And they may still miss some flows or some packets in the flows. To illustrate the effect of our cocktail approach, we first use *M1+M2* methods to identify the T1~T5 traces. And then we use cocktail approach (*M1+M2+M3*) to identify these traces again. The update interval is 8 minutes.

A. Recall Rates

Fig.6 shows the recall rates in trace T1~T5. Our cocktail approach can achieve very high recall rate, no matter what identification level is concerned. At the level of flows (R_f), cocktail approach can identify 80%~90% BT flows, while the *M1+M2* only identify 22%~39% flows. But at the level of packets and bytes, it looks like that *M1+M2* can achieve better performance. The recall rate of packets and bytes, R_p and R_b, are between 55%~72% and 70%~86%. While R_p and R_b of our approach are still between 85%~95% and 88%~98%. The reason lies in those un- or semi- established BT flows. For BT clients, after they obtain the list of peers they will try to connect the remote peers with the TCP handshake and then BT handshake or MSE handshake. Among these flows, part of them may have only a *SYN* packet and will not get any response for all sorts of reasons. And another part of them may establish the TCP connections, but can not establish BT or MSE connections. Thus in these flows, there are no application signatures or any MSE features. Thus the *M1+M2* will miss them, which are also missed by most of existing approaches. While our approach, especially *M3*, can foresee these BT flows by analyzing the signaling. So we will not miss them, although there are few packets or bytes belong to them. Therefore, no matter what identification level is concerned, our approach is stable.

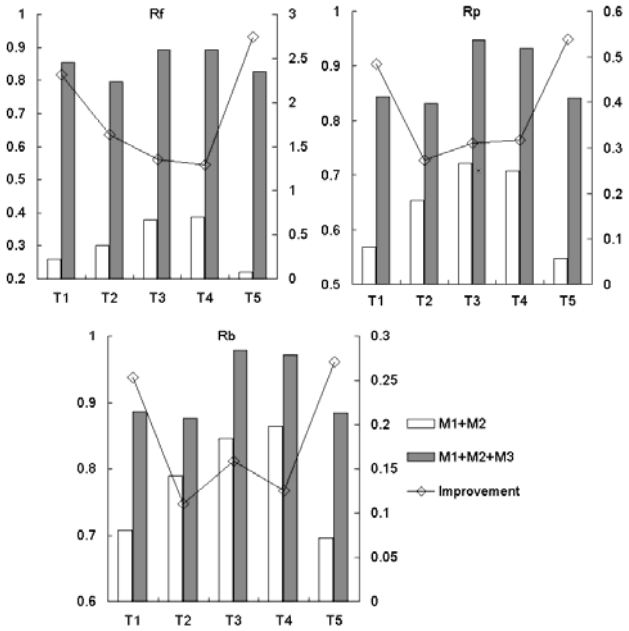


Figure 6. Recall Rates

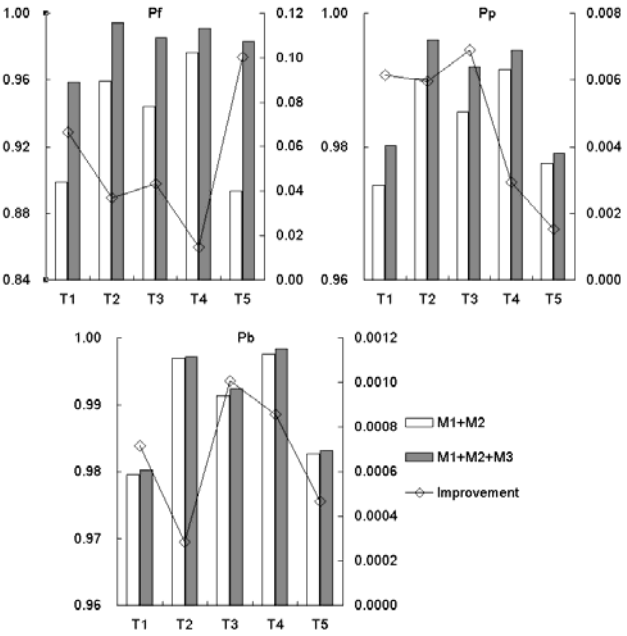


Figure 7. Precision Rates

B. Precision Rates

Fig.7 gives the data of precision rates. It looks like a draw between cocktail approach and *M1+M2*, especially at the level of bytes. The *M1+M2* have achieved high precision rates already. The P_f is between 89.35~97.65%, the average is 93.44%. The P_p and P_b are between 97.42~99.16% (98.37%) and 97.95~99.69% (98.96%) separately. While the P_f, P_p and P_b of cocktail approach are 95.87~99.46% (98.26%), 97.90~99.60% (98.83%) and 98.02~99.84% (99.03%) separately.

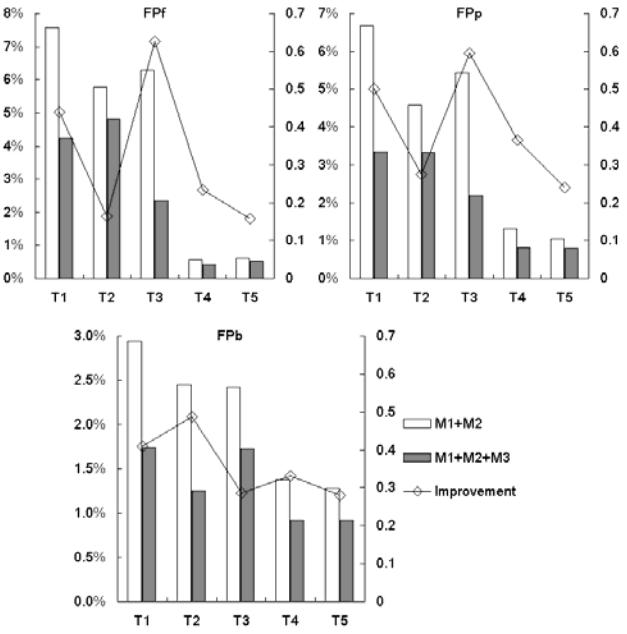


Figure 8. False Positive Rates

C. False Positive Rates

Fig.8 gives the data of false positive rates. And we can find that on all traces our approach can efficiently decrease the false positive rates. The FP_f of *M1+M2* on five traces is between 7.58~0.56%, the average is 4.16%. The FP_p and FP_b are between 6.68~1.05% (3.81%) and 2.94~1.28% (2.09%) separately. While the FP_f, FP_p and FP_b of cocktail approach can drop to 4.81~0.43% (2.47%), 3.34~0.80% (2.10%) and 1.74~0.92% (1.31%) separately. But our approach still introduces few false positives which are caused by HTTPS and SMB (Server Message Block) protocols. It is easy to distinguish these false positive flows by port-mapping, because they are running on fixed port of 443 and 445.

D. Real-time Identification

Now, we evaluate our approach the ability of real-time identification, as shown in Fig. 9 and Fig. 10.

In Fig. 9, we find almost 90% flows can be identified at the first packet, which is the TCP *SYN* packet. Because most upcoming flows are predicted by *M3* with the list of peers extracting from the signaling packets. And another peak appears at the point of 4 on x-axis. It means about 5% flows can be identified at the fourth packet. They are unencrypted BT flows, and the *M1* match the signatures in the first BT handshake packet after 3-way TCP handshake, which is the fourth packet of that flow. The remaining flows are left to *M2*. Most of them can be identified before the twelfth packet. So if a flow can not be identified in 12 packets, it has few chances, less than 2%, to be a BT flow. Thus, the weighted average number of packets should be checked by cocktail approach before the flow is determined as BT flow or not is 1.66.

If we consider the time before a flow can be identified, we find that over 85% flows can be identified in 100 ms, even in 1ms, as shown in Fig. 10. That is to say they are identified as soon as they are generated. This all contributes to the *M3* by wiretap the signaling. The most of remaining flows will be identified in 1000 ms by *M1* and *M2* together. And another interesting phenomenon is that the handshake process of some flows may continue a relative long time before the packet containing the signatures is transmitted or two peers finish the MSE handshake. It looks like a long tail. To identifying these flows, we must wait and store the necessary information of them, such as five tuples of flows, which occupy the resource of CPU and memory. It will greatly degrade the real-time identification ability of our approach. So if a flow can not be determined in 1000 ms, it should be omitted or treated as none BT traffic. But it will cause little false negatives.

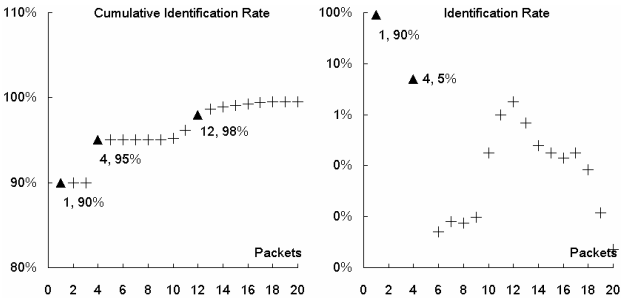


Figure 9. Identification cost on packets

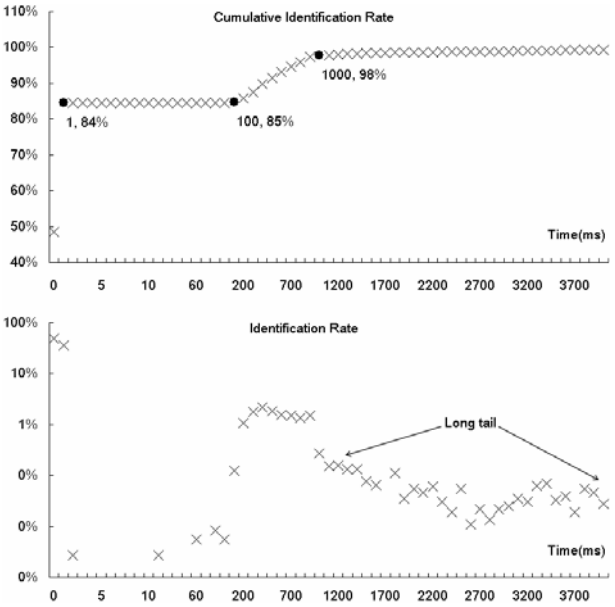


Figure 10. Identification cost on time

VI. Conclusions And Future works

P2P traffic identification is a hot spot of research in recent years. Lots of creative approaches are proposed from different viewpoints. But the real-time identification is still a hard task. In this work, we a cocktail approach consists of three sub-methods to identify BT traffic in real-time. These sub-methods combine with each other and aim at identifying different kinds of BT traffic at the very beginning of or even before the traffic generated. Results indicate that near 84% flows can be identified at the first packet and in 1ms. And to evaluate our approach, we use modified Vuze clients, not only to generate real BT traffic, but also to label out these traffic in benchmark traces by themselves. It can reach very high accuracy and help us to well evaluate our cocktail approach. Thus this cocktail approach achieves the ability of real-time identification with high accuracy, low overhead.

And next, we plan to model the BT signaling process to reduce the overhead our approach further.

Acknowledgment

We would like to thank the Suzhou Key Laboratory of Converged Communications for the assistance in deploying experiment environment. And this work is supported in part by the projects of Suzhou Science and Technology Department under grant No.SZS0805, No.ZXG0825, SYJG09024 and SYG201034.

References

[1] IPOQUE. Internet Study 2008/2009. *http://www.ipoque.com/resources/ internet-studies/ internet-study-2008_2009.*

[2] Z, Yang, L.Z. Li, Z.H. Wang and L. Q. Li. Active analysis of BT with a modified Azureus client, in proceedings of the *10th IEEE International Conference on Computer and Information Technology*, Bradford, UK, 2010, pp. 371-376.

[3] S. Sen and J. Wang, Analyzing peer-to-peer traffic across large networks, *Transactions on Networking*, 2004, 12(2), pp.219-232.
[4] A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen, P2P The Gorilla in the Cable, in *National Cable &Telecommunications Association 2003 National Show*, Chicago, IL, 2003.
[5] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, An Analysis of Internet Content Delivery Systems, in proceedings of the *5th Symposium on Operating Systems Design and Implementation*, Boston, MA, 2002, pp. 315-328.
[6] A.W. Moore, K. Papagiannaki, Toward the Accurate Identification of Network Applications, in proceedings of the *6th International Workshop on Passive and Active Measurement*, Boston, MA, 2005, pp. 41-54.
[7] S. Sen, O. Spatscheck, D. Wang, Accurate, scalable in-network identification of P2P traffic using application signatures, in proceedings of the *13th international conference on World Wide Web*, New York, USA, 2004, pp. 512-521.
[8] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos, Is P2P dying or just hiding? in proceeding of the *47th annual IEEE Global Telecommunications Conference*, Dallas, USA , 2004, pp. 1532-1538.
[9] X. Lu, H. Duan, X. Li, Identification of P2P traffic based on the content redistribution characteristic, in proceedings of the *International Symposium on Communications and Information Technologies*, Sydney, NSW, 2007, pp. 596-601
[10] P. Haffner, S. Sen, O. Spatscheck, D. Wang, ACAS: automated construction of application signatures, in proceedings of the *ACM SIGCOMM Workshop on Mining Network Data*, New York, USA, 2005, pp. 197-202.
[11] K. Xu et al., Identify P2P traffic by inspecting data transfer behavior, *Computer Communications*. 2010, 33(10), pp.1141-1150
[12] J. Ma, K. Levchenko, C. Kreibich, S. Savage, G.M. Voelker, Unexpected means of protocol inference, in: proceedings of the *6th ACM SIGCOMM Conference on Internet Measurement*, New York, USA, 2006, pp. 313-326.
[13] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, Transport layer identification of P2P traffic, in proceedings the *4th ACM SIGCOMM Conference on Internet Measurement*, Taormina, Italy, 2004, pp. 121-134.
[14] A. Moore and D. Zuev, Internet traffic classification using Bayesian analysis, in proceedings of the *2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Alberta, Canada, 2005, pp. 50-60.
[15] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification, in proceedings of the *4th ACM SIGCOMM Conference on Internet Measurement*, Taormina, Italy, 2004, pp. 135-148.
[16] N. Williams, S. Zander, G. Armitages, A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification, *SIGCOMM Computer Communication Review*, 2006, 36 (5), pp.5-16.
[17] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, C. Williamson, Offline/realtime traffic classification using semi-supervised learning, *Performance Evaluation*, 2007, 64 (9-12), pp. 1194-1213.
[18] L. Bernaille, R. Teixeira, K. Salamatian, Early application identification, in proceedings of the *2nd ACM Conference on Future Networking Technologies*, Lisboa, Portugal, 2006, pp. 436-455.
[19] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, P. Tofanelli, Revealing skype traffic: when randomness plays with you, in proceedings of the *SIGCOMM 2007*, Kyoto, Japan. 2007, pp. 37-48
[20] M. Crotti, F. Gringoli, P. Pelosato, L. Salgarelli, A statistical approach to ip-level classification of network traffic, in proceedings of the *IEEE International Conference on Communications*, Istanbul, Turkey, 2006, pp. 170-176.
[21] F. H. Campos, F.D. Smith, K. Jeffay, A.B. Nobel. Statistical clustering of internet communication patterns, in proceedings of the *35th Symposium on the Interface of Computing Science and Statistics*, Salt Lake City, UT, 2003, pp. 134
[22] A. McGregor, M. Hall, P. Lorier, J. Brunskill, Flow clustering using machine learning techniques, in proceedings of the *5th International Workshop on Passive and Active Network Measurement*, Antibes Juan-les-Pins, France, 2004, pp. 205-214.
[23] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, BLINC: multilevel traffic classification in the dark, in proceedings of the *ACM SIGCOMM 2005*, Philadelphia, PA, 2005, pp. 229-240.
[24] F. Constantinou and P. Mavrommatis, Identifying Known and Unknown Peer-to-Peer Traffic, in proceedings of the *5th IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, 2006, pp. 93-102.
[25] K. Xu, Z.L. Zhang, S. Bhattacharyya, Profiling internet backbone traffic: behavior models and applications, in proceedings of the *ACM SIGCOMM 2005*, New York, USA, 2005, pp. 169-180.
[26] Yan Hu , Dah-Ming Chiu, John C.S. Lui. Profiling and identification of P2P traffic. *Computer Networks*, 2009, 53 (6), pp. 849-863
[27] J. Li, S. Zhang, Y. Lu, and J. Yan, Real-Time P2P Traffic Identification, in proceedings of the *49th annual IEEE Global Telecommunications Conference*, New Orleans, LA, 2008, pp. 1-5.
[28] J. Ma, K. Levchenko, C. Krebich, S. Savage, and G. Voelker, Unexpected Means of Protocol Inference, in proceedings of the *6th ACM SIGCOMM Conference on Internet Measurement*, Rio de Janeiro, Brazil, 2006, pp. 313-326.
[29] B. Cohen. The BitTorrent Protocol Specification. *http://www.bittorrent. org/beps/bep_0003.html*
[30] MSE: Message stream encryption protocol. *http://www.azureuswiki.com/index.php/MessageStreamEncryption*.
[31] Diffie, W., Hellman, M.E.: New directions in cryptography. *IEEE Trans. Information Theory IT*, 1976, 22(6), pp. 644-654
[32] Thayer, R., Kaukonen, K. A stream cipher encryption algorithm 'arcfour'. *http://www.mozilla.org/projects/security/pki/nss/draft-kaukonen-cipher-arcfour-03.txt.*
[33] Peer Exchange. *http://wiki.vuze.com/w/Peer_Exchange.*
[34] V. Paxson, Bro: a system for detecting network intruders in real-time, *Computer Networks*, 1999, 31(23), pp. 2435-2463.
[35] Z. Yang, L. Z. Li, Q. J. Ji, Y. Q. Zhu. An omnibus identification of BitTorrent traffic in a stub network, in proceedings of the *3rd International Symposium on Parallel Architectures, Algorithms and Programming*, Dalian, China, 2010, pp. 346-353.

Zhe Yang was born in Suzhou, Jiangsu province, in 1978. He received the B.S. degree in computer application from Donghua University, Shanghai, in 2000. He received the M.S. and Ph.D degree in computer application technology form Tongji University, Shanghai, in 2003 and 2006.

He is working in Soochow University. And his research interests include traffic classification, P2P behavior analysis and network measurement.

Ling-Zhi Li was born in Dezhou, Shandong province, in 1977. He received the B.S. and M.S. degrees in information technology from East China Institute of Technology, Fuzhou, in 1998 and 2001. He received the Ph.D. degree in computer application from Nanjing University of Aeronautics and Astronautics, Nanjing, in 2006.

He is working in Soochow University. His research interests include traffic engineering, network security.

Qijin Ji was born in Anhui province China, 1974. He received the M.S degree in information engineering from Nanjing University of Posts and Telecommunications, Nanjing, China and Ph.D degree in computer science and engineering from Southeast University, Nanjing, China, in 2002 and 2005 respectively.

After a two year experience as a Post-doctor fellow in Shanghai Jiaotong University, Shanghai China, he joined the School of Computer Science and Engineering of Soochow University and currently he is an assistant professor. His research focuses on performance evaluation and resource control of computer networks, distributed estimation and decision making in multi-agent systems and distributed multimedia streaming systems.

Yanqin Zhu was born in Wuxi, Jiangsu province, in 1964. She received the B.S. degree in Mathematics from Soochow University in 1985，the M.S. degree in computer application technology form Southeast University in 1991, and Ph.D degree in computer application technology form Soochow University in 2008.

She has been working in Soochow University since 1985. And her research interests include computer network and applied cryptography.

A Context-Aware Routing Protocol on Internet of Things Based on Sea Computing Model

Zhikui Chen, Haozhe Wang, Yang Liu, Fanyu Bu, Zhe Wei
School of Software, Dalian University of Technology
Dalian 116620, China
zkchen@dlut.edu.cn, wang.haozhe0@gmail.com

***Abstract*— The paper syncretizes the fundamental concept of the Sea Computing model in Internet of Things and the routing protocol of the wireless sensor network, and proposes a new routing protocol CASCR (Context-Awareness in Sea Computing Routing Protocol) for Internet of Things, based on context-awareness which belongs to the key technologies of Internet of Things. Furthermore, the paper describes the details on the protocol in the work flow, data structure and quantitative algorithm and so on. Finally, the simulation is given to analyze the work performance of the protocol CASCR. Theoretical analysis and experiment verify that CASCR has higher energy efficient and longer lifetime than the congeneric protocols. The paper enriches the theoretical foundation and makes some contribution for wireless sensor network transiting to Internet of Things in this research phase.**

***Index Terms*— context-awareness, Internet of Things, routing protocol, energy efficient, Sea Computing**

I. INTRODUCTION

Recent rapid progress in sensor, networking, and RFID (Radio Frequency Identification Devices) technologies allows connecting various physical world objects to the cyber infrastructure, which could enable realization of the Internet of Things (IOT) vision. IOT is a technological revolution that represents the future of computing and communications, and it aims at increasing the ubiquity of the Internet by integrating every object for interaction via embedded systems, which leads to the highly distributed network of devices communicating with human beings as well as other devices[2]. Sea Computing is a new model of IOT, which emphasizes the intelligence of the nodes. Interaction between nodes has been significantly enhanced, and the nodes deal with the gathering information themselves rather than transmit it to the others [3].

The technologies of wireless sensor network (WSN) are the concrete representations for implementing the IOT vision, and the specialties low-dissipation, low-cost, distributed and self-organized bring an important revolution in the computer application domain [4]. The essential characteristic of the IOT is transparency. The context-aware computing will debase the degree of the humankind participating straight greatly then implementing transparency intercommunion through apperceiving the useful information to reason, decide and calculate automatically. Therefore, the research on the context-aware computing is very important to implement the imagination of the IOT. Its appearance and development correspond to the need of individual and multi-kind in the future with high informationization[5-6]. This paper chooses the context-awareness to be the research keystone as the junction of the IOT thought and WSN technology. The protocol CASCR (**C**ontext-**A**wareness in **S**ea **C**omputing **R**outing Protocol) is brought forward in this paper on aiming at the existent problems in the existing WSN routing protocol. Some promoting effect reference value and scientific basis to some extent will be brought in the domain of the IOT.

The rest of the paper is organized as follows. In Section II, we discuss the related work. In Section III and IV, we present the key technologies, context-awareness and sea computing, of the paper. Section V describes our proposed routing protocol for IOT with formal definitions. We evaluate our protocols in Section VI, and conclude in Section VII.

II. RELATED WORK

The academia sorts the IOT/WSN routing protocols some kinds according to different standards [7-8]. It can be sorted into two kinds below by the communication ways of the nodes: complanate routing protocols and hierarchical ones. Concretely, the hierarchical ones need the support of the basal general routing protocol working in the clusters. They also can be sorted into the ones based on the data and the ones based on the location information by the discovering procedure of the routing.

The IOT/WSN routing protocols are not mature enough, and lots of them are in the academic research phase. Some analysis to the classical ones will be given as below [9-10].

(1) The protocol LEACH (Low Energy Adaptive Clustering Hierarchy) ignores the concrete energy problem that the cluster head nodes assemble together easily in the election so as to the illusive randomicity. The formula considers the randomicity and misses the logicality.

(2) Although SPIN (Sensor Protocols for Information via Negotiation) solves the resource wasting like information blast and information redundancy through the negotiating mechanism with the localized topology as the CASCR in this paper, the complicated “three times shake-hands” and the inflexible energy threshold all make SPIN poor in the energy efficient.

doi:10.4304/jcp.7.1.96-105

(3) As the representation of the routing protocol based on the query, the manner Rumor on the random single-cast lack the guarantee in reliability and logicality. Rumor cannot get rid of the unoptimizable problem and the routing ring in the path producing.

(4) The routing algorithm EPGR (Energy Prediction and Geographical Routing) given in the reference [11] cannot reflect most WSN nodes' different working states practically with the node's states designed in it. And its designing is complicated and poorly representational; The algorithm DMMDR (Dynamic Multifactor Markov Decision Routing) given in the reference [12] designs the sensor nodes into different states, but the topology variety's uncertain and frequency in the practical deployment of WSN will make the scale of the "states" here instable and uncertain. The DMMDR discusses the WSN topology problem with the topological structure only and is lack of the application rationality; the model CTMPC (Context-based Triggered Task Model in Pervasive Computing) given in the reference [13] is an application representation of the context-aware computing. But this model and other similar research thoughts given by the academia are only settling in the theoretical reasoning level and the qualitative plan level. They are unable to deploy quantificationally and standardly in the concrete application as the superficial research level of the context-aware computing. The application of the context-awareness in the pervasive computing environment in the future must merge with mathematics tool so that the integrated indices of the WSN routing technology can be improved better; the algorithm MAFZP (Mobility Aware Fast Zoned Protocol) given in the reference [5] does not make full use of the context-aware computing with the far-fetched designing model. It cannot reflect the context relationship of the nodes' movement well and be lack of the technical interfusion in the details. So it cannot be considered as a representative designing of the context-awareness core thought merging with the concrete application deeply for the lower specialty exertion of the context-aware computing.

III. CONTEXT-AWARE COMPUTING

What is called context-awareness is a kind of controlling, obtaining and analyzing technology to the context information. And the mutative information of this ambient environment is so-called "context", for example, the geography location, environment information, jitter data or states of the monitored target.

The context-aware computing includes five sub-technologies mainly: (1) context-getting (2) context-modeling (3) context-reasoning (4) context-conflict-solving (5) context-storage and management.

A context-aware system means that a system which can use the context information and do some corresponding change or configuration automatically as the variety of the context information to provide the users individual proper context service [13]. The protocol CASCR given in this paper makes the IOT a context-aware system.

The context-awareness has an extensive application field in human production living like the intelligent official work, house-hold service, medical treatment and so on. For implementing the application system intelligent, controllable and predictable highly as the final destination of the context-aware computing, making the topology states in the IOT system predictable and reasonable to some extent through utilizing the IOT nodes' relevant real-time information, data, states and parameters as the context knowledge is positive to the IOT routing technology in the present development phase [14-15].

IV. SEA COMPUTING MODEL

Sea Computing is a new computing model of IOT. Through embodying the computing unit and communications equipment in the physical world objects, the objects can interconnect with each other. Even in the scene which is unpredictable in advance to judge, Sea Computing can realize the interaction between nodes. Its essence is making the information device invisibly integrate into the real physical world everywhere, and extending the informatization to the physical world.

On the one hand, by strengthening the variety embodying information devices in the objects, making objects and information devices achieve a close integration, the objects could natively obtain the information about the physical world; on the other hand, by strengthening the local real-time interaction and distributed intelligence between mass individual things, the objects will be provided with self-organization, self-calculation and self-feedback of Sea Computing functions. The goal of Sea Computing is the intelligent communication between objects, realizing the interaction of things, emphasizing the intelligent connection and physical properties emergence in the physical world. It is a physical world-centric perspective.

The characteristic of Sea Computing is [3]:

(1) **Embodiment**. Information devices are embodied into various things, so do the sensors. The information devices have the same lifecycle as the object, and it is self-management and self-maintenance.

(2) **Autonomy**. The objects are not controlled passively, but possess a certain autonomous and autonomic ability.

(3) **Local Interaction**. Sea Computing make full use of locality principle, and things are mainly through local interaction to realize communication.

(4) **Swarm Intelligence**. Sea Computing model of IOT is a dynamic self-organizing system. The intelligent algorithm in one node can't know the result in advance, and it needs to interact with the other ones through the embedded intelligent algorithm to produce the effective intelligent decision.

Due to the characteristic described above, in Sea Computing model the routing protocol not only transmits the data, but also perceives the data. The routing is established to transfer the data, and meanwhile, the data promotes the routing. CASCR based on Sea Computing uses context-aware computing to perceive the information and to optimize the routing.

V. THE PROTOCOL CASCR

With syncretizing the context-aware computing, the protocol CASCR (Context-Awareness in Sea Computing Routing Protocol) considers the inherent specialties of the IOT adequately and utilizes multi-kinds context information to describe and review the work state data of the nodes roundly. It can control the nodes' variety trend in the long time running effectively and elevate the using efficiency of the energy.

A. Formal Definitions and Basal Work flow &Thought

In this section we will define some fundamental thoughts of CASCR protocol, and the detail information will discuss in the next section.

For the brevity of discussion, we use the following notation for formal definitions and quantitative.

$N(A)$	node with identifier A
$Nei(A)$	neighbor nodes of $N(A)$
$Sup(A)$	superior nodes of $N(A)$
$Sub(A)$	subordinate nodes of $N(A)$
$Col(A)$	colleague nodes of $N(A)$
$S(A)_{f-w}$	one of the five states for $N(A)$ represents that $N(A)$ is in the state of full-working, and from this state, $N(A)$ can change into the other states under some specific condition
$S(A)_{ser}$	the second state which means $N(A)$ is in the state of serving
$S(A)_{s-w}$	the third state, which means $N(A)$ is in the state of single-working
$S(A)_{sle}$	the forth state, which means $N(A)$ is in the state of sleeping
$S(A)_{hib}$	the fifth state, which represents that $N(A)$ is in the state of hibernating
$slice(n)$	the n_{th} time unit
$Rate(M)_{Energy-Consumption}$	dynamic context information of $N(M)$, which represents the rate of energy consumption in the past timeslices
$Instant(M)_{Consumption-Rate}$	dynamic context represents the instant value of consumption of $N(M)$
$Life(M)_{\mathrm{Re}sidual-Energy}$	dynamic context represents the lifetime of residual energy of $N(M)$
$Queue(M)_{Pending-Tasks}$	dynamic context which stands for the pending task queue of $N(M)$ in the future timeslices
$CF(E_i)$	trust degree for the state transforming probability
$Dis(A,B)$	the number of hops between $N(A)$ and $N(B)$
$N(A) \lhd M(X)$	$N(A)$ receives message M from $N(X)$
$WT(A)$	the working time of $N(A)$
H_j	transferring the current state to state j
EC	short for $Rate(M)_{Energy-Consumption}$
CR	short for $Instant(M)_{Consumption-Rate}$
RE	short for $Life(M)_{\mathrm{Re}sidual-Energy}$
PT	short for $Queue(M)_{Pending-Tasks}$
$Tran(A)$	the transferring process of $N(A)$, working with H_j to specify the node's ID
$CF_{H_j}(E)$	the trust degree for state transforming to state j
$P_{H_j}(E/S)$	the posterior probability of trust degree for transforming to state j, being equivalent to $CF_{H_j}(E)$
$P(H_j/S)$	the posterior probability of H_j, equivalent to $P_{H_j}(E/S)$

The complanate routing protocol is more efficient than the hierarchical ones in the middle or small scale IOT network and inside the clusters. It has the low complexity, implementation cost, good controllability, transplantation and extensive application range characteristic. So the CASCR given in this paper designed as a complanate routing protocol can be transplanted into the clusters of the hierarchical routing protocol.

Each node has a relative notation set with the other nodes, which are $Nei(A)$, $Sup(A)$, $Sub(A)$ and $Col(A)$. For the further discussion briefly we define them first.

Definition 1: (1) The neighbor nodes of $N(A)$, $Nei(A)$, is a set of nodes which can reach node A with only one hop, i.e., $Nei(A) := \{N(x) \mid x : Dis(N(x), N(A)) = 1\}$.

(2) The superior nodes of $N(A)$, $Sup(A)$, is a set of neighbor nodes which send the gathering message to N(A), i.e., $Sup(A) := \{N(x) \mid x : N(x) \in Nei(A), N(A) \lhd M(x)\}$.

(3) The subordinate nodes of $N(A)$, $Sub(A)$, is a set of neighbor nodes which receive the message from node A, i.e., $Sub(A) := \{N(x) \mid x : N(x) \in Nei(A), N(x) \lhd M(A)\}$.

(4) The colleague nodes of $N(A)$, $Col(A)$, is a set of neighbor nodes which neither are the $Sup(A)$ nor the $Sub(A)$, i.e.,

$$Col(A) := \{N(x) \mid x : N(x) \in Nei(A), N(x) \notin Sup(A), N(x) \notin Sub(A)\}$$

Figure 1 illustrates one example of the relationships of the nodes. The nodes $N(0)$, $N(5)$, $N(3)$, $N(1)$, $N(2)$ and $N(7)$ are all the neighbor nodes of $N(6)$, Therein, the $N(0)$, $N(5)$ and $N(3)$ are the superior nodes of the $N(6)$; As the same, $N(7)$ is the subordinate node; $N(1)$ and $N(2)$ are the colleague nodes of $N(6)$. But $N(4)$ is not like any one of them.

The subordinate and superior nodes can exchange in Sea Computing model, because in IOT a node can transmit the gathering data to the subordinate nodes, and also can send the controlling message to the superior nodes. But in this paper, we make a unified definition to prevent misleading.

There are some different work schemes in the sensor nodes to suit the different work environment and the mutative portfolio flow with time. Every work scheme predefined is called a work state. The nodes adopt different work manners and energy scheme to achieve the goal that adapt the environment and save energy through changing their work states. There are five different states defined in the protocol CASCR.

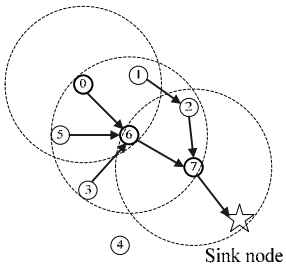


Figure 1. The relationships of the neighbor, superior, subordinate and colleague nodes

Definition 2: Operations and States

Operations:

Operation 1, gathering: The node's sensor assembles the information from the surrounding environment.

Operation 2, transmitting: The node's transceiver transfers the data and controlling operations for its superior and subordinate nodes.

Operation 3, fusing: The node's intelligent information device deal with the gathering data by compressing and fusing before transmitting them to other nodes.

Operation 4, controlling: The node receives the operation order, and the intelligent algorithm of information device control the object.

Operation 5, Sleeping: Through the intelligent algorithm of information device, the node could turn into sleeping. If the node is sleeping, it will not transmit the data, and the processor runs in very low power. But if the node needs to come back for work, the transceiver and processor will launch quickly.

Operation 6, Hibernating: The node shuts itself down, and it would not gather data, transmit information or fuse it. But the timer is still working, so that the node still synchronizes with the others.

States:

State 1: $S(A)_{f-w}$, which means that every module in the node A is in the working state. The node needs to do the former four operations above including gathering, fusing, and sending the controlling data that needs to be done by itself and at the same time transmits the data information of its superior nodes.

State 2: $S(A)_{ser}$, which means the node accomplishes its work and transmits the data and controlling operations for its superior.

State 3: $S(A)_{s-w}$, which means the node just gathers the environment information and fuses it by itself but not any other nodes.

State 4: $S(A)_{sle}$, which means that the transmitter will be shut down and the processor will be down to the lowest. The receiver will be set to work on the low power waste state.

State 5: $S(A)_{hib}$, which means that the transmitter, receiver and processor will be all shut down except the timer. The node stays in the absolute still state.

Definition 3: The long time running nodes in the WSN will work as the timeslice that are equal in length. So the transition of the states in the Definition 2 is according to the timeslice as basal units. The sensor nodes must work in a kind of states n slice time. Thereinto, n is integer.

N(A)'s working time can include one or more timeslice. $WT(A) := \{\{slice(t)\} \mid t \in \mathbb{N}^k\}$, $\mathbb{N}$ is infinite integer domain.

Definition 4: The dynamic context of the nodes is the most straight and effective context information that can be researched quantificationally. The dynamic context of the nodes are the information in all kinds that can be used to describe the dynamic work status of the nodes in the IOT application environment including the energy consuming rate, processing task quantity per fixed time, and the work states' variety information of the nodes and so on.

The context data message for election to the next hop just transmit among the nodes that have straight operation contact i.e. the operation superior just calculating and forecasting the future work state for its operation subordinate and then the useful context information in single hop area will be in control in the IOT which running the CASCR. The whole IOT network has initialized steady effective routing before the CASCR's running. Based on this precondition, we consider that

there are not redundant route, roundabout route and short-inefficient ones in any local topology of the IOT. And some trivial simple questions will not be given definitely for simplifying the length of this paper.

B. Detail Definition and Data Structure

There are five states: full-working, serving, single-working, sleeping, hibernating states defined in the CASCR as the program in the section above, definition 2. Before discussing the details, we first introduce one rule for the nodes' working states transferring.

Rule: The node cannot transfer to the state 4 or state 5 from the state 4 or state 5, i.e.,

$$Tran(A) := \{H_j \mid j : S(A)_{sle} \to \neg H_4, S(A)_{sle} \to \neg H_5, S(A)_{hib} \to \neg H_4, S(A)_{hib} \to \neg H_5, j \in [1,5]\}$$

The state 4 and 5 can only be the transition results of the other three states. This design will decrease the probability of false disability of the vast nodes in the IOT.

The protocol CASCR combines with the AI (artificial intelligence) theory and Markov probability tool quantificationally process the data in kinds gathered by the nodes at every time slice according to a set of adjustable states transition rules and then input the quantitative module of the CASCR for the effective parameters that can be used to forecast the states variety trend at last. Like the figure 2 below.

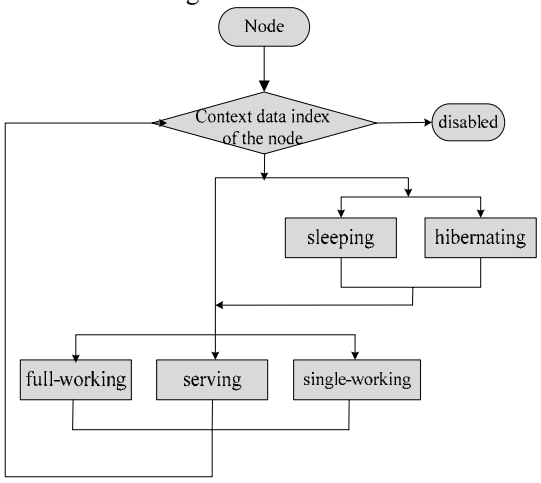


Figure 2. The demonstration of the transitions between the states of the nodes

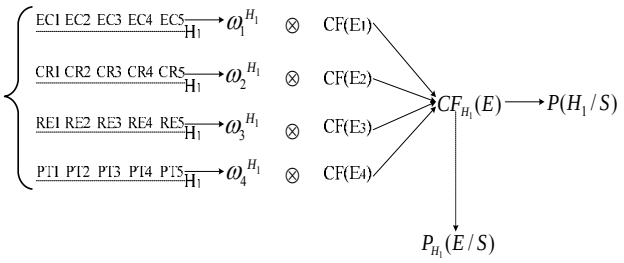


Figure 3. The processing thought on the main data of the CASCR

Through utilizing the 5×4 data as sets (illustrating by figure 4) to compute with the adjustable states transition conclusion $H_1 \sim H_5$ stored in the nodes, the weight $\omega_i^{H_j}$ relative to every conclusion will be obtained and then the weights will combine with trust degree $CF(E_i)$ of that kind of the data, i.e. the trust degree of the evidence in this observation. After that, the integrated accordant degree $CF_{H_i}(E)$, i.e., $P_{H_i}(E/S)$, relative to conclusion H_j for the node in this timeslice will be obtained. It is considered as posterior probability $P(H_1/S)$ of the integrated evidence in the dynamic observation. At last, the state transition matrix stored in the node will be initialized through computing to get other four posterior probability values to the other four states. These values are all based on the context data in a timeslice's first half. In this paper, a time slice unit is set as 10 minutes to be an example (the concrete value can be adjusted in the different scenes or simulation environment). So the first 5 minutes in the timeslice will represent the node's working state in the second 5 minutes and even the future.

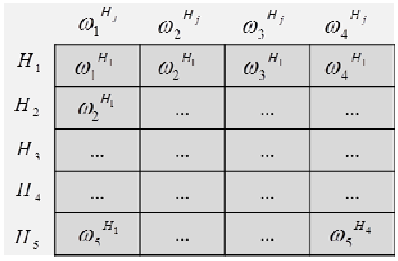


Figure 4. The mapping for $\omega_i^{H_j}$ stored in node

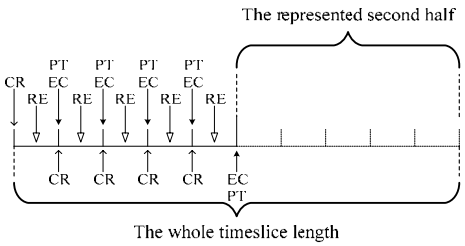


Figure 5. Collect the context data in a time slice

Definition 5: Rate of the node's energy consumption in the i_{th} timeslice, $Rate^i_{Energy-Consumption} = \frac{e_{i+1} - e_i}{1}(\mathrm{J/min})$, there into e_i is the residual energy value of the battery in node as the unit Joule, and the positive integer i is in the field [0,4].

Gather five residual energy values of the battery a, b, c, d, e as the unit Joule and use the latter value minus the former to get the real energy consumption in the past 1 minute. For the six values: $e_1 = a$, $e_2 = b$, $e_3 = c$, $e_4 = d$,

$e_5 = e$ and one initial value $e_0 = ini$, there are five $Rate^i_{Energy-Consumption} = \frac{e_{i+1} - e_i}{1}$ values. These five parameters will be used as $Rate(M)_{Energy-Consumption}$ data.

Definition 6: Instantaneous value of consumption, $Instant(M)_{Consumption-Rate}$, is obtained through gathering instantaneous energy consumption rate value on the five times instant and use them as $Instant(M)_{Consumption-Rate}$ directly with the unit $^{J/\min}$.

Definition 7: Queue of pending tasks data, $Queue(M)_{Pending-Tasks}$, is obtaining in the same way as the CR data. Gather five residual data queue length values of the node and use them as the $Queue(M)_{Pending-Tasks}$ directly. The measurement of the queue length is according to the concrete IOT environment with the fixed length data as the standard value $^{100\%}$.

Definition 8: Lifetime of residual energy, $Life^i_{Residual-Energy} = \frac{\frac{e_{i+1}}{CR^{i+1}} - \frac{e_i}{CR^i}}{\frac{e_5}{CR^5}}$, thereinto e_i is the residual energy value of the battery in node as the unit Joule, and the positive integer i is in the field [1,4]; CR^i presents instantaneous energy consumption rate $Instant^i_{Consumption-Rate}$.

The RE data is for gathering five residual time data of the battery. Divide the present residual energy value e_i by the present instantaneous energy consumption rate value z_i and five instantaneous residual time value of the battery $t_i = \frac{e_i}{z_i}$ will be obtained with the positive integer i in the field [1,5]. We use the latter value minus the former to get the residual time rate of change of the battery relative to the one minute past for reflecting the degree of change and consumption rate of the battery energy residual time in this first half time slice. It can be positive or negative. Four v_i values will be got through $v_i = t_{i+1} - t_i$ with the positive integer $i \in [1,4]$. At last, we divide v_i by the last value t_5 to get four $LifeTime_{Residual-Energy}$ data, i.e. $LifeTime^i_{Residual\ Energy} = \frac{v_i}{t_5}$. Several types of the four data in the matching with the transition rules (refer to the Table 1); the semantic distance method will not be adopted for the unidirectivity of the data bound in the items. It is judged just by how many data accord with the standard values in the four. It will be set to 1 if all do with also can be set to $\frac{3}{4} \times 1$ if there are three do. The other items will be calculated by the ways of semantic distance.

The protocol CASCR chooses the next hop succedaneous node in the single hop area. The superior nodes send a routing election request message in their neighbor area and the destinations will be confirmed as the environment context data table stored in the nodes (illustrating by Figure 6 and 7). The destination nodes that receive the request message will reply a routing election reply message according to their context data situation (illustrating by Figure 8). CASCR don't adopt the manner of periodic synchronization to decrease the cost of communication and quantity of the packets among the nodes.

Routing Election Request Message			
1	2	3	4
Dest	Src	Next hop	Second hop

Figure 6. The format of the Routing Election Request message

Environment Context Data Table			
Present neighbor node	…	…	…
Present subordinate node	…	…	…
Present superior node	…	…	…
Present colleague node	…	…	…
temporary disabled node	…	…	…

Figure 7. The structure of the Environment Context Data Table

The third and fourth item in the Figure 8 are used to judge whether the reply nodes are the neighbour of the nodes will be replaced and its next hop one.

Several data structures stored in the nodes are the context data table (CDT) to store the context data information of this node, the subordinate context data table (SCDT) to store context data information of the subordinate nodes of this node, the environment context data table (ECDT) to store context data information of the ambient nodes of this node in a single hop neighbour area. The traditional routing table has the problems of items scale, time to live, onefold function and out of date and so on [16]. The CDT implements the functions of the traditional routing table in the conventional routing protocol uniformly and exceeds the category of the onefold routing information.

Routing Election Reply Message																													
1	2	3	4	5						6						7						8						9	
Dest	Src	Boolean1	Boolean2	$Rate_{Energy-Consumption}$						$Instant_{Consumption-Rate}$						$LifeTime_{Residual-Energy}$						$Queue_{Pending-Tasks}$						$CurrentState$	
ID1	ID2	Y/N	Y/N	EC1	EC2	EC3	EC4	EC5	CF(E1)	CR1	CR2	CR3	CR4	CR5	CF(E2)	RE1	RE2	RE3	RE4	RE5	CF(E3)	PT1	PT2	PT3	PT4	PT5	CF(E4)	1/2/3/4/5	

Figure 8. The format of the Routing Election Reply message

Context Data Table																							
$Rate_{Energy-Consumption}$						$Instant_{Consumption-Rate}$						$LifeTime_{Residual-Energy}$						$Queue_{Pending-Tasks}$						*CurrentState*
EC1	EC2	EC3	EC4	EC5	CF(E1)	CR1	CR2	CR3	CR4	CR5	CF(E2)	RE1	RE2	RE3	RE4	RE5	CF(E3)	PT1	PT2	PT3	PT4	PT5	CF(E4)	1/2/3/4/5

Figure 9. The structure of the Context Data Table in this node

Subordinate Context Data Table					
Node ID	$Rate_{Energy-Consumption}$	$Instant_{Consumption-Rate}$	$LifeTime_{Residual-Energy}$	$Queue_{Pending-Tasks}$	*CurrentState*
ID1	…	…	…	…	…
ID2	…	…	…	…	…
…	…	…	…	…	…
…	…	…	…	…	…

Figure 10. The structure of the Subordinate Context Data Table

C. Quantitative Module of CASCR

The context data gathered on the time slice are processed according to the 5 states transition rules to get the weights values $\omega_i^{H_j}$ that represent the influence of these context data toward the states' transition respectively. We use the formula fuzzy match theory, Minkowski distance, like equation (1) [17]:

$$d(A,B)=[\frac{1}{n}\times\sum_{i=1}^{n}\left|\mu_A(u_i)-\mu_B(u_i)\right|^{\sigma}]^{1/\sigma}\quad(\sigma\geq 1) \tag{1}$$

A relevant standard transition value in an item of the transition rules is defined as S. So we measure the relevant parameters which represent the target node in the first half time slice with the S as the standard. For example, the x_1, x_2, x_3, x_4, and $U=\{u_1,u_2,u_3,u_4\}=\{x_1,x_2,x_3,x_4\}$ as the universe of discourse then $A=\{\mu_A(u_i)|u_i\in U, i=1,2,3\cdots\}$, $B=\{\mu_B(u_i)|u_i=S,\quad i=1,2,3\cdots\}=\{1,1,1,1\}$, thereinto, the fuzzy set A represents the degree that u_i approaches to S. The membership function designed in the CASCR is as the formula (2), and it is easy to prove that formula (2) obey a normal distribution.

$$\mu_A(u_i)=\mathrm{e}^{-\lambda(u_i-S)^2} \tag{2}$$

$CF(E_i)$ is the key parameters for evaluating a set of evidence's trust degree. An array of data from the reality which represent the states of the nodes to some extent need to be measured by the way of the degree of the jitter of this kind of data in some range. We will turn this set of data down in the trust degree if the jitter extent of the data exceeds our practice to the WSN's work situation, otherwise reversely. This mechanism restricts the relevant weights that participate in the reviewing to the CASCR effectively. The jitter extent of the sample data will be mapped to the relevant trust degree $CF(E_i)$, i.e. the probability determinacy of this set of data evidence in this samples gathered observation. The CASCR utilizes cloud model theory and improves it proper for the trust degree computing [18-20]. The formula (3), (4), (5),(6) are as below.

$$\widehat{En}=\sqrt{\frac{\pi}{2}}\times\frac{1}{n}\sum_{i=1}^{n}\left|X_i-\widehat{E(X)}\right| \tag{3}$$

$$\widehat{E(X)}=\frac{1}{n}\sum_{i=1}^{n}X_i \tag{4}$$

$$\widehat{He}=\sqrt{S^2-\widehat{En}^2} \tag{5}$$

$$CF(E_i)=\mathrm{e}^{-\varepsilon\widehat{He}} \tag{6}$$

Thereinto, X_i is the single sample observation value, with S^2 the sample variance and $\widehat{E(X)}$ is sample average, i.e., $\widehat{E(X)}=\overline{X}$. We use the formula (7) to merge the weights of the multi-kinds sample data with the relevant trust degree to obtain $P_{H_j}(E/S)$.

$$P(H_j/S)=P_{H_j}(E/S)=$$
$$CF_{H_j}(E)=\frac{1}{\sum_{i=1}^{n}\omega_i^{H_j}}\sum_{i=1}^{n}(\omega_i^{H_j}\times CF(E_i)) \tag{7}$$

The table of the states transition rules of the nodes is stored in the nodes as Table I.

At last, we forecast the target nodes' transition trend in the future time slice through Markov probability tool. With the equation Chapman-Kolmogorov [21]:

$$P_{ij}(u+v)=\sum_{k=1}^{\infty}P_{ik}(u)P_{kj}(v),\qquad i,j=1,2\cdots \tag{8}$$

Because the node's state transition rule is matched through fuzzy formula, it means that the next timeslice state is only relative with current situation. So the state transition is a Markov process. We use the relationship $\mathrm{P}(n)=\mathrm{P}(0)^n$ to calculate the 5×5 state transition matrix.

The results of probability calculated by C-K equation are stored in the Probabilistic or-set table (p-or-set tables for short) [22]. We define the p-or-set table in another way, which makes it more suitable for our algorithm.

N_1	N_2	N_3	N_4	...
$\langle H^1_{1,1}:P^1_{1,1},H^1_{1,2}:P^1_{1,2},H^1_{1,3}:P^1_{1,3},H^1_{1,4}:P^1_{1,4},H^1_{1,5}:P^1_{1,5}\rangle$	$\langle H^1_{2,1}:P^1_{2,1},H^1_{2,2}:P^1_{2,2},H^1_{2,3}:P^1_{2,3},H^1_{2,4}:P^1_{2,4},H^1_{2,5}:P^1_{2,5}\rangle$	$\langle H^1_{3,1}:P^1_{3,1},H^1_{3,2}:P^1_{3,2},H^1_{3,3}:P^1_{3,3},...\rangle$	$\langle ... \rangle$	...
$\langle H^2_{1,1}:P^2_{1,1},H^2_{1,2}:P^2_{1,2},H^2_{1,3}:P^2_{1,3},H^2_{1,4}:P^2_{1,4},H^2_{1,5}:P^2_{1,5}\rangle$	$\langle H^2_{2,1}:P^2_{2,1},H^2_{2,2}:P^2_{2,2},H^2_{2,3}:P^2_{2,3},H^2_{2,4}:P^2_{2,4},H^2_{2,5}:P^2_{2,5}\rangle$	$\langle ... \rangle$	$\langle ... \rangle$	...
...	...	...	...	...
...	$\langle \prod_{k=1}^{5} (H^i_{j,k}:P^i_{j,k}) \rangle$	...	...	...

Figure 11. Probabilistic or-set-table for CASCR

TABLE I. STATES TRANSITION RULES OF THE NODES

EC	*CR*	*RE*	*PT*	*Conclusion*
＜0.5	≤3	＞-0.5	＜3	H1: transfer to state 1
＜0.5	≤3	＞-0.5	0	H2: transfer to state 2
0.5～1	3～4	-0.5～-1	＞3	H3: transfer to state 3
1～2	4～5	-1～-2	0～5% or ＞6	H4: transfer to state 4
＞2	＞5	＜-2	0 or ＞8	H5: transfer to state 5

Definition 9: A p-or-set tables is a probability space in which the set of outcomes $\mathbb{S}$ is finite, and whose outcomes are all the probabilistic counterpart, i.e.,

a pair $(\mathbb{S}, p)$ *where* $\sum_{H\in\mathbb{S}} p(H)=1$. Therein, $\mathbb{S} := \{H_j \mid j \in [1,5]\}$.

In each p-or-set table cell the pair is defined as

$$\langle \prod_{k=1}^{5} (H^i_{j,k}:P^i_{j,k})\rangle = \langle H^i_{j,1}:P^i_{j,1}, H^i_{j,2}:P^i_{j,2}, H^i_{j,3}:P^i_{j,3}, H^i_{j,4}:P^i_{j,4}, H^i_{j,5}:P^i_{j,5}\rangle,$$

therein, i is the number of rows whereas j is the number of columns, and k presents the state number.

Fig. 11 illustrates the p-or-set table for CASCR. The column number depends on the set size of $Nei(A)$, and the table cells are the probabilistic counterpart where the attribute values are the state transition H_j, and plain probability. In the first cell of Row 2, the content means N_1 has $P^1_{1,1}$ probably transit to State 1, and $P^1_{1,2}$ probably transit to state 2. We can consider the p-or-set table as the Markov state transition matrix, and the number of rows presents the number of order of the transition matrix.

VI. SIMULATION AND RESULT

This paper designs the virtual environment of the simulation experiment based on the physical equipment. It is built according to the real data of the sensor nodes. The CASCR is compared with the classical LEACH and SPT(shortest path tree) at the aspects of the number of the alive nodes, average energy dissipation and so on. The simulation tool is Matlab 7.8.0 and the environment parameters are in the Table II below.

TABLE II. THE PARAMETERS IN THE SIMULATION ENVIRONMENT

Parameters	*Values*
area acreage	1500m×1500m
source voltage	3V(DC)
transmitting current	30mA
transmitting power	+0.5dBm
receiving current	37mA
single hop distance	150m
antenna type	omni antenna
number of nodes	50 to 100
power upply capacity/node	60mAh
λ	0.0078
σ	2
ε	0.4621
timeslice	60s
standard queue	10K
data rate of node	250Kb/s
contrastive protocols	LEACH & SPT

The simulation work is comparing with the classical LEACH of the hierarchical routing and the classical SPT of the complanate routing.

As illustrated in Figure 12, we run the protocol with 100 nodes lasting 50 minutes. The survival time of nodes which running CASCR is longer than the other two protocols. In the beginning, CASCR will collect the context information from the neighbor nodes, and that would cause some nodes to consume more energy. So there are some nodes die early in the first 5 minutes. But when the nodes get adequate context information, the CASCR algorithm will be self-optimizing, hence the lifetime of CASCR nodes are two and a half times of SPT and one and a half time of LEACH.

From 10th minute, nodes start to die, and the LEACH and SPT's alive nodes number sharply decline. However, CASCR uses context-ware technology to maintain alive nodes number, and we can see the number of alive nodes running CASCR decline slowly.

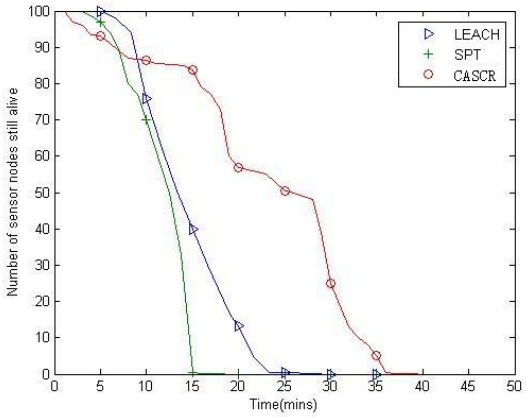


Figure 12. The number of the nodes still alive with time varying in the three protocols

Figure 13 shows the average energy dissipation of the three protocols. From the results, we can see CASCR also has better energy consumption control than the other two. As the discussion above, CASCR's nodes need to collect context information first and calculate through Markov, so its energy consumption in the first 10 minutes is more than LEACH and SPT. But from 10th minute, the energy consumption of CASCR is much lower, which is in accord with the alive node number.

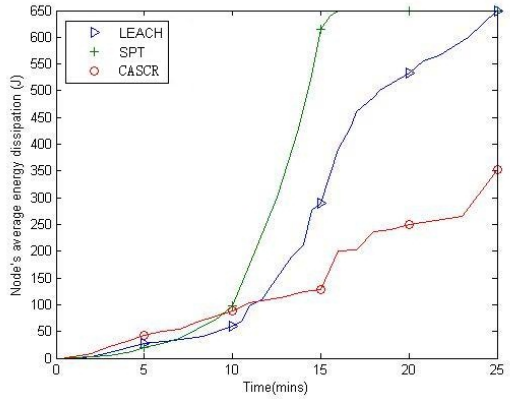


Figure 13. The node's average energy dissipation with time varying in the three protocols

Figure 14 is a comparison in total quantity of the gathered data with the nodes' number varying. We run the three protocols for 20 minutes with 2 nodes increased every time, and the nodes number is increased from 50 to 100. SPT is suited to the small scale network. When the scale of nodes is small, for example less than 60 nodes, the SPT performs the best. However, with the number of nodes increased, SPT's performance descends radically. Since LEACH is a hierarchical routing protocol, when the nodes' scale becomes larger, the performance increases along. For the CASCR, the quantity of the gathered data is one weakness, because in CASCR the nodes have 5 states, including sleeping and hibernating, some nodes may not gather data all the time. But as illustrated in Figure 12, the lifetime of CASCR is 1.5 times of LEACH, and the average quantity of data gathered by LEACH is 1.39 times of CASCR, due to CASCR with long lifetime, for long-time running CASCR will have a better performance.

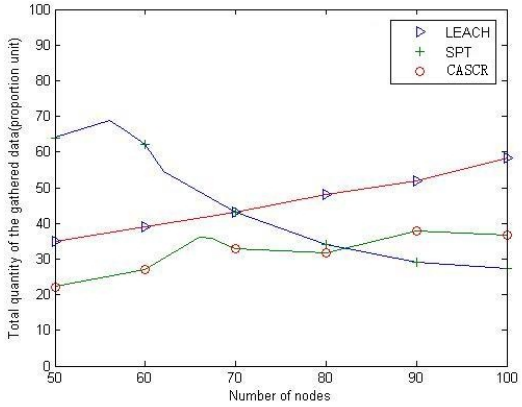


Figure 14. The total quantity of the gathered data with the nodes' number varying in the three protocols

Figure 15 shows the control packet statistic with the number of nodes varying in the three protocols. We run the protocols for 20 minutes with 2 nodes increased every time, and the nodes number is increased from 50 to 100. As illustrated in the figure, the control packet statistic of SPT is always the least, because once the shortest path tree is built, it would not send much control packets. LEACH is a hierarchical routing protocol, and it will always cycle randomly pick the cluster-head node then broadcast the head node's information. So with the increase in the number of nodes, LEACH will send more control packets. For CASCR, the control packets are used to multicast the dynamic context information and the multicast frequency is every half time slice. CASCR is in the middle level, better than LEACH and little worse than SPT. However, SPT's lifetime is short and energy dissipation is high, which makes its performance of it is not good like the other two.

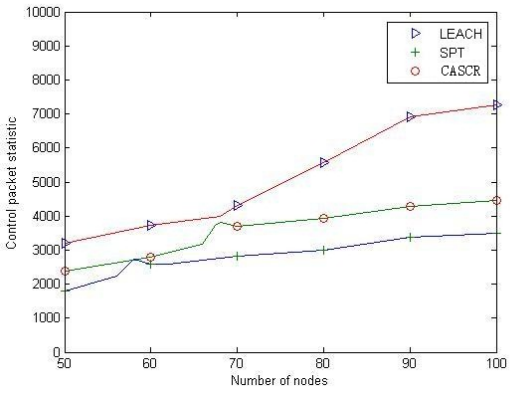


Figure 15. The control packet statistic with the nodes' number varying in the three protocols

The result shows that the CASCR has a better performance to some extent than the other two protocols.

VII. CONCLUSION

In an IOT network, nodes have stronger sensory ability and smarter intelligence. So in this paper we

propose the protocol CASCR as the representative which using the core idea of the context-aware computing into the basis routing protocol. Context information will become more and more important with the development of IOT, because we will not satisfy by gathering the data, but knowing the meaning of the data.

Based on the Sea Computing model, we transform the traditional routing thought into a positive predictable routing protocol with using fuzzy match theory of artificial intelligence and Markov probability model, which contain some forwarding and directive meaning. It is an exploratory evolution direction of the WSN technology in the accordance with the IOT guidance feature in the future.

Through the simulation and analysis of CASCR, the results show that the CASCR has longer life time and lower energy consumption, which indicated that CASCR is more suitable for the IOT routing based on Sea Computing model. But it still has abundant space for improving and the core protocol CASCR proposed here can be improved continuously by the theory circles as a fundamental protocol framework, and can also be enhanced to a cluster-type protocol. We will keep on doing the research on this field.

ACKNOWLEDGMENT

The work reported in this paper has been partially supported by the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Zhe Wei, Zhikui Chen, Yang Liu et al. “A WSN Routing Protocol Based on the Context-Aware Technology”. International Symposium on Parallel Architectures, Algorithms and Programming, China, 2010.
[2] International Telecommunication Union. “ITU Internet Reports 2005: The Internet of Things”. 7th edition, 2005.
[3] Sun Ninghui, Xu zhiwei, Li guojie. “Sea Computing: a New Computing Model of Internet of Things”. Communications of the CCF,6(7):52-57, 2010.
[4] Wang Shu, Yan Yujie, Hu Fuping et al.”The Theory and Application of the Wirless Sensor Networks”. Beijing:the Beihang University Press,2007.
[5] Li Rui. ”Research on Some Key Techniques of Context-Aware Computing”. Changsha: Hunan University, 2007.
[6] Kirankumar B. Balavalad, S.S. Manvi, A.V. Sutagundar. “Context aware computing in wireless sensor networks”. International Conference on Advances in Recent Technologies in Communication and Computing, 2009.
[7] Stankovic, J.A.. “Wireless Sensor Networks”. IEEE Computer Society,41(10):92-95,2008.
[8] Tang Yong, Zhou Mingtian, Zhang Xin. “Overview of Routing Protocols in Wireless Sensor Networks”. Journal of Software, 17(03):410-421, 2006.
[9] Yu Hongyi, Li Ou, Zhang Xiaoyi et al.Wireless Sensor Networks Theory, Technique and Implementaion. Beijing:National Defence Industry Press,2008.
[10] Shen Bo, Zhang Shiyong, Zhong Yiping. “Cluster-Based Routing Protocols for Wireless Sensor Networks”. Journal of Software,17(07):1588-1600, 2006.
[11] Lin Kai, Zhao Hai, Yin Zhenyu et al.Energy prediction and routing algorithm in wireless sensor network.Journal on Communications,27(05):21-27, 2006.
[12] Liu Tiantian, Jia Zhiping. “Dynamic Multifactor-Based Markov Decision Routing in Embedded Communication System”. Journal of Shanghai Jiaotong University,41(11):1816-1819,2007.
[13] Luo Junwei, Qing Xiao, Chen Sigong. “Context Based Triggered Task Model in Pervasive Computing”. Mini-micro Systems, 25(08):1542-1545, 2004.
[14] Xi Min, Qi Yong, Zhao Jizhong. “Prediction Based Load Balance Algorithm in Context-aware Application Server”. Computer Engineering,34(03):124-126, 2008.
[15] Li Caixia, Zhou Jiliang, Cao Qiying et al. “Research on location prediction based on context”. Application Research of Computers, 25(11):3325-3327, 2008.
[16] Chen Linxing, Zeng Xi, Cao Yi. “Mobile Ad hoc:Self-organizing packet Wireless Networks Technique”. Beijing:Electrical Industry Press,2006.
[17] Wang Yongqing. “Artificial Intelligence principle and methods”. Xi'an: Xi'an Jiaotong University Press,2006.
[18] Li Deyi, Meng Haijun, Shi Xuemei. “Membership Clouds and Membership Cloud Generators”. Journal of Computer Research and Development, 32(06):15-20,2006.
[19] Li Deyi, Liu Changyu, Du Yi. “Artificial Intelligence with Uncertainty”. Journal of Software,15(11):1583-1594,2004.
[20] Chen Zhikui, Zi Bingjie, Jiang Guohai et al. “Trust evaluation for wireless sensor networks based on trust-cloud”. Journal of Computer Applications, 30(12):3346-3348, 2010.
[21] Sheng Zhou, Xie Shiqian, Pan Chengyi. ”Probability and Mathematical Statistics 4th Edition”. Beijing:Higher Education Press,2008
[22] Green TJ, Tannen V.. “Models for incomplete and probabilistic information”. IEEE Date Engineering Bulletin, 29(1): 17-24, 2006.

Zhikui Chen is a professor and PhD supervisor of Dalian University of Technology. He is also the director of institute of network communication and database, director of joint research institute of Dalian university of technology and Going.com of Tokyo. He has published more than 80 refereed papers in relevant journals and conferences and he has received national and international funding for projects related to the above research areas.

Haozhe Wang is a graduate student in Dalian University of Technology and his research interests are related to data processing of context-awareness and data fusion in wireless sensor networks and Internet of Things.

Yang Liu is a PhD student in Dalian University of Technology and his research interests are related to Cognitive Radio and Information Security.

Fanyu Bu is a PhD student in Dalian University of Technology and his research interests are related to Internet of Things and wireless sensor networks.

Zhe Wei is a graduate student in Dalian University of Technology and his research interests are related to Context-awareness computing in Wireless Sensor Networks

An Energy-Aware Multi-Core Scheduler based on Generalized Tit-For-Tat Cooperative Game

Guowei Wu
School of Software, Dalian University of Technology, Dalian, China
E-mail: wgwdut@dlut.edu.cn

Zichuan Xu, Qiufen Xia, Jiankang Ren
School of Software, Dalian University of Technology, Dalian, China
E-mail: {eggerxu, xiaqiufen, rjk.dlut}@gmail.com

***Abstract*—Energy-constrained computing environments are emerging those years, especially in embedding computing. A game theoretic energy-aware scheduling algorithm for multi-core systems is proposed in this paper, namely, GTFTES (Generalized Tit-For-Tat Energy-aware Scheduling). GTFTES is designed to work in a resource-rich environment where resources always compete for tasks. A generalized Tit-for-Tat based method, where whether a core will cooperate or not is decided by a hardness factor, is considered in this paper. The algorithm is implemented in our EASS simulator. Simulations results show that the proposed game can reduce the temperature difference between different groups of cores which effectively avoids the local hotspot of a processor.**

***Index Terms*—energy-aware scheduling, multi-core, game theory, generalized tit-for-tat**

I. INTRODUCTION

Resources in embedded systems such as power and computation capacity are often limited in both industrial and mobile applications. It is deserved attention that there exists an increasing trend in the use of multi-core processor in energy-constrained embedded and mobile systems, where real-time guarantee must be met without exceeding safe temperature levels within the processor. However, multi-cores increase power consumption which in turn creates temperature problems and non uniform power density map. As high-performance embedded systems become increasingly energy-constrained, the question of how to balance the energy distribution against real-time guarantee must be addressed.

Dynamic Thermal Management (DTM) techniques are the primary solutions to lower energy consumption in hardware level. Clock gating and Dynamic Voltage Frequency Scaling are two reactive DTM mechanisms. When the core power rises to a critical level, clock gating technique switches off the core clock for a certain period of time and then switches it on. During this time the core is slowing down the total processing time and in turn decreases power consumption. Dynamic Voltage Frequency Scaling (DVFS) decreases the frequency and voltage of processors at run time to achieve lower power consumption. Hardware level thermal management can respond rapidly when the processor is overheated. However, it has no information on the behavior of the applications when the processor is running, i.e., hardware level thermal management always responds in the same way independent from the workload, moreover, hardware level thermal management mechanism increases chip cost.

Recently, software level energy management technique has been studied because of its low cost. The primary advantage of software technique is that it can obtain the information of running threads; hence it can reduce the processor energy or temperature according to the characteristics of the thread. One primary approach adopted in operating system level for energy management is energy aware scheduling, which takes processor power into account to avoid thermal violation, and keeps the temperature below a certain limitation by throttling the "hot" tasks. Authors in [1-3] present algorithms for thread migration. However, they do not take real-time characteristics into consideration. Paper [4] proposes a thread scheduling scheme for a class-based 2D Mesh interconnected multi-core system which considers inter-core distances in its scheduling decisions. Without considering the temperature property, their algorithm is hard to fulfill the thermal constraints. Authors [5] only analyze the relationship between activity migration and power density. Some thermal-aware scheduling algorithms are only suitable for CMP architecture [5-7]. Algorithms in [8] aims to minimize the energy consumption and the makespan of complex computationally intensive scientific problems, subject to energy constraints. Moreover, authors in [6] also did some research the problem of power-aware scheduling/mapping of tasks by applying game theory in [9]. However, their algorithms only consider energies

Corresponding author: Zichuan Xu
Email:eggerxu@gmail.com

doi:10.4304/jcp.7.1.106-115

rather than temperatures. In contrast, we propose a more detailed and energy-aware scheduling environments where players (cores) are competing for the tasks, and from a more realistic perspective, temperatures are used to evaluate the algorithm's performance.

The problem we consider falls under the class of resource allocation problems. Although these problems have been exploded theoretically and practically in areas such as gird computing, scheduling and wireless network [10-12], it's still a novel approach to apply market models into the constraint-aware scheduling, such as power-aware, temperature-aware scheduling. In the most common form of auctions, the highest bidder wins the resource and pays as much as the bid. However, the users may have an incentive to lie about their true value of the resource. A Vickrey auction [13] is a type of sealed-bid auction, where bidders submit written bids without knowing the bid of the other people in the auction. The highest bidder wins, but the price paid is the second-highest bid. This type of auction gives bidders an incentive to bid their true value.

Above lists common game theoretic methods which works in a non-corporative and Tit-for-Tat manner. Yet, Tit-for-Tat, as traditionally defined, makes sense only for a two-player game. For a many core processor, the cores in it are always highly interacted especially in many core systems. For example, if one core is thermally saturated, and its neighbor cores are often influenced for the heat transferring. Thus, it is reasonable to model the scheduling by utilizing a corporative game model. In this paper, we adopt the generalized Tit-for-Tat mechanism to model the scheduling problem. The objective is to minimize power density of the processor and temperature of each core.

This paper is an extension in terms of both theoretic and simulation results of our conference paper [14]. The main contributions of this paper are as follows.

- A generalized tit-for-tat based corporative scheduling game is proposed in the energy-aware scheduling algorithm.
- A simple but feasible and efficient temperature calculation method is presented.
- Extensive simulations on our energy-aware scheduling simulator, namely EASS, are given.

The rest of the paper is organized as follows. In section II, we present some notations and the problem definition. In section III, the GTFTES algorithm and temperature calculation algorithm is stated in this section. In section IV, the GTFTES algorithm is implemented in EASS and compared with other related algorithms. The experimental results and performance analysis is listed in this section. In section V, we summarize related work. We conclude the paper in Section VI.

II. PRELIMINARIES

In this section, we give some notations and the definition of the problem. We introduce the scheduling environment considered by this paper and the concepts of game theory.

A. A Scheduling Environment

This paper adopts a resource-rich scheduling environment in which numerous execution cores complete the tasks. The reason that we adopt such a circumstance is that, currently, the execution cores of commercial CPU, especially GPU are soaring in an astonishing rate. It is common to see hundreds of cores in a GPU. However, since not all users' daily tasks can be related to GPU, there is circumstances that majority of the execution units of GPU are idle, which also means number of tasks is less than the cores .

A multi-core scheduling environment (shown in Figure 1) in which cores are competing for tasks is proposed in this section. In such environment each core wants to maximize its income through executing tasks. Tit-for-Tat is a highly effective strategy in game theory for the iterated prisoner's dilemma. In this paper, we consider a generalized tit-for-tat method where each player decide to cooperate or retaliate according to the hardness factor h. Specifically, after each round of the game, each player will calculate the proportion that the players who cooperate (this proportion is specified as hardness h). If h is over a predefined value, the player will decide to cooperate in the next game round. Otherwise, he will choose to retaliate.

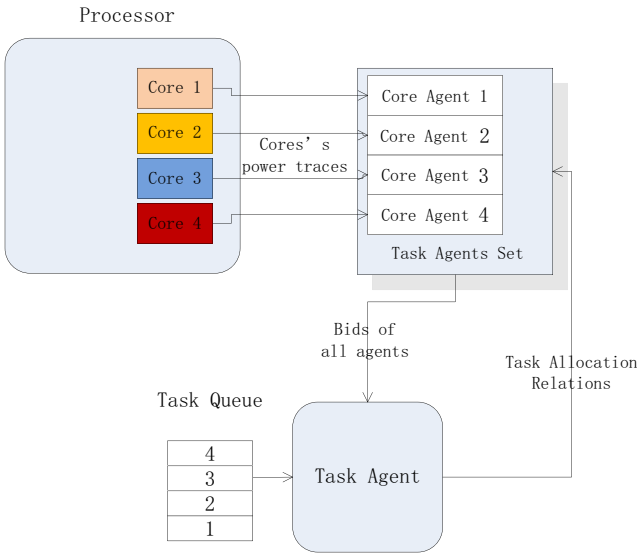


Figure 1. Scheduling Environment of GTFETS

Auctioning is a common way of allocating tasks to cores. In an auction each core bids a certain amount to buy a task. As the cores are selling its services, we use a reverse second price auction.

We consider a more general and realistic scenario where cores may have potential to cooperate to avoid the hotspot and reduce the energy consumption of a processor. And also, they are rational which means they will choose to retaliate

without considering the power status of the processor. Consider a energy-constrained embedded system that consists of a set of tasks, $T = \{\tau_1, \tau_2, \ldots, \tau_i\}$ $(0 < i < N)$ and a set of execution cores $EC = \{ec_1, ec_2, \ldots, ec_j\}$.

To state and analysis the algorithm, we need some assumptions which are stated next.

A.1 No task is starved of wealth.

A.2 The scheduling environment is resource-rich. As in market, this is a model of supply exceeds demand. Thus, with this assumption, the cores compete for the tasks sounds reasonable.

Then, the symbols used in this paper are listed in Table I.

TABLE I.
SYMBOLS AND NOTATIONS

Symbol	Notation
ec_i	The execution core on which the current thread is running
w_i	The overall wealth that a core bids at each time slice.
b_i	The core ec_i 's bid
c_j	The jth execution core where $0 < j \le M$
C_c	Currently available cores.
T_c	Temperature set of the currently available cores $T_c = \{t_j\}$
p_j	The price that core ec_j has to pay for the winning of an auction
g	The biding function of each task.
h_{th}^i	The core's hardness threshold which is actually a percentage of plays who choose to cooperate in a new round of game.
S_{th}	The temperature threshold of a execution core.
T	The set of tasks that are ready to be executed.
γ	The weighting coefficient that a agent to evaluate the value of a core
L_{avg}	The average execution time of a task on a core
φ	The weighting coefficient on L_{avg} when task agents are calculating the bids

B. Game Theory and Generalized Tit-for-Tat

Game theory is a multi-player decision theory. The players of a Game are subjects that make decisions. The players participate in a Game in order to get maximum benefit by selecting a reasonable action. Thus the elements of a Game include players, information, strategy spaces and payoff functions. In our scheduling environment, the competition of task j is a Game among N cores. Game theory involves three basic elements, which is defined as follows: Player set N : The set of all execution cores. $EC = \{1, 2, \ldots n\}$. Strategic set S_i : For each core $i \in N$ owns a strategic set, which comprises several strategies $S_i = \{s_i\} = \{s_1^{(i)}, s_2^{(i)}, \ldots, s_n^{(i)}\}, i = 1, 2, \ldots, n$. After each core determines its strategy, then $s = \{s_1, s_2, \ldots, s_n\}$ expresses action of the N cores at one time. We utilize s_i to represent the ith core's strategy, whereas s_{-i} is the strategy set of all the other cores. Payoff function $u_i : S \to R$, representing the benefits of cores in different strategies. Hence each cores wishes to seek for the maximal profit; therefore it intends to enlarge the benefit of its payoff function.

Generally, in game theory, effectiveness of a strategy is measured under the assumption that each player cares only about him or herself. By this game-theory definition of effectiveness tit for tat was superior to a variety of alternative strategies. However, in reality, tit-for-tat is only applicable to two player game where non-corporative actions are common. Generalized tit-for-tat is an extended version of tit-for-tat which could be used in the multi-player game. In this mechanism, in each auction, players can decide to cooperate or retaliate according to their observations.

III. GAME THEORETIC TEMPERATURE-AWARE SCHEDULING ALGORITHM

In this section, a game theoretic thread allocation in the energy-constrained systems and a temperature calculation method which is used in the simulator to evaluate the effectiveness of the GTFETS algorithm are proposed. Moreover, the theoretic results of the task allocation game are analyzed.

A. Problem Definition

We consider a scheduling environment with an energy-limited processor executing tasks whose number is always lower than cores (as assumption **A.2** shows). Thus, a scenario of cores competing for different tasks is considered in this paper.

In reality, the cores favor to get a task to execute for the sake of getting as much personal profit as possible. Actually, they only consider their own payoffs rather than the power status of a processor. However, for multi-player games in economic environments, there do exist some corporations where a group of players have a social goal. The 'HotSpot' of a processor may cause instability of the processor and even hardware damage [15], which is a reason that the processor performs to achieve a more uniform thermal status of itself for the sake of longer servicing time. In our previous work [16], we proposed temperature-aware scheduling algorithm which the temperature is calculated in a pre-set interval. Though, by adopting this method, the hotspots of processors can be efficiently eliminated, it is more expensive to do so than just using the power which can be calculated from the voltage. Thus, we consider energy factor as the main social goal of cores, and treat the temperature as a measurement of the algorithm.

One reverse VCG auction is actually one type of the sealed bid auction where the resource could be divided to be allocated to the tasks. In our scheduling scenario, each core ec_i submits a price that a task should pay for the executing service. Naturally, the core biding the lowest bids will win the auction. The auctioneer (the scheduler proposed in this paper) firstly decides the winning cores, and then assigns the tasks to the cores. As what it does in VCG, the auctioneer charges each core the harm they cause to other cores, and ensures that the optimal strategy for a task is to bid the value of the cores.

Formally, this N-player game can be written as $G=\left[N,\{S_i\},\{u_i(\cdot)\}\right]$ which specifies for each player i a set of strategies, or bidding functions, S_i (with $s_i \in S_i$) and a payoff function $u_i(s_1,\cdots,s_n)$ giving the computational capability associated with outcome of the auction.

B. Task Allocation Algorithm

As described in section 3.1, we consider the auction happens at the beginning at each time slice. However, in practical applications, not all the cores are willing to choose to cooperate in an auction for the sake of social goal. Also, the core affinity of each task is an important issue that a task allocation algorithm should consider. In the GTFTES algorithm, the task preferentially choose its previous execution if it is in the winners of each auction. The task agent's algorithm is listed in Table II.

TABLE II.
TASK AGENT'S ALGORITHM

Algorithm Name	Task's Algorithm
Input:	Bids set $B_i=\{b_i\}$, ready set of tasks T
Output:	The task allocation relation
1.	Rank the bids in non-increasing order. The winners are the top $\lvert T\rvert$ cores.
2.	Calculate the winners' payment. A winner's payment is the least bid of the winner set without its participation.
3.	If the previous execution core of task is in the winner set and it is currently not ocupied. Allocate itself to its previous core.
4.	Else, allocate the tasks whose previous execution core is not in the set of winners according to the tasks contribution. Specifically, the tasks with lower contribution will be allocated to a core with lower bid.

TABLE III.
CORE AGENT'S ALGORITHM IN ROUND K

Algorithm Name	Cores' Algorithm
Input:	The core's hardness threshold h_{th}^i and temperature threshold S_{th}, the set of tasks T
Output:	The core's bid vector b_i.
1.	Carry out the valuation of itself and calculate its thermal status S_i.
2.	if $S_i > S_{th}$, the core is forced to cooperate in this round of auction. And, calculate its bid by its thermal status, that is, $b_i=\gamma\cdot P_i$. Then jump to step 6.
3.	Calculate the hardness h_k in the previous round of the auction, which aims to decide whether to cooperate or not.
4.	if $h_k > h_{th}^i$, the core will choose to cooperate in this round of auction. And, calculate its bid by its power status, that is, $b_i=\gamma\cdot P_i$.
5.	if $h_k < h_{th}^i$, the core will choose to retaliate. And, calculate its bid by the average execution time of the tasks in T, that is, $b_i=\varphi\cdot L_{avg}$.
6.	Send the bid to auctioneer (Task Agent).

The core agent's algorithm in the auction of each time slice is to carry out a proper bid. Firstly, the cores should decide whether to cooperate or not. If corporation is selected, the core has to carry out for a social goal—more uniform power status of the processor. Specifically, if the core has a higher power consumption in the passing ten time slices, it should carry out a much higher bid to avoid being selected to execute a task. The algorithm is listed in Table III.

C. Temperature Calculation Method

Real temperature measure method, like integrating advanced thermometers, can undoubted get very precise temperature dynamics. However, that kind of equipments are not integrated in all systems, or the number of them is limited. Some models calculate the temperatures by completely theoretic methods, which may be so unrealistic to be utilized to real systems. In this section we use an integrated method of calculating temperatures which utilized the empirical temperature property of an instruction-level event.

The GTFETS algorithm is evaluated by the temperatures traces of the processor in the simulation, and the calculation method is presented in this subsection. The symbols used in the temperature calculation method are listed as follows.

- T_p: the temperature of the processor (°C).
- $q_j(x,y)$: the power density of the jth core (W/m^2).
- $u_j(x,y,t)$: the temperature of the jth core on plane z at time t
- S_j: the area of core j
- γ_j: the constant values that used in the superposition of different cores' temperature.
- h: the conductance of processor's interface material, $h=k/d$.
- a_v: event power, which is counted by performance counter v.
- t_j: the number of times the event j happens.

We adopt the similar thermal model used in [17] and its physical architecture is depicted in Figure 2.

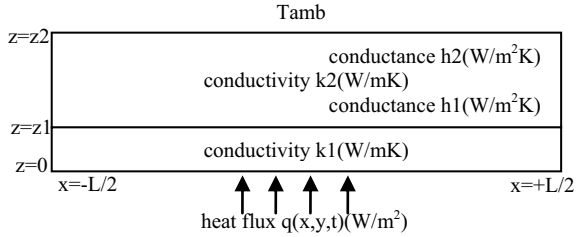


Figure 2. ATMI physical model

Let $u_{(0,0,0)}(x,y,t)$ to denote the temperature generated on the plane $z=0$ when a core located at point $(0,0,0)$. Then the temperature of the *jth* core is:

$$u_j(x_j,y_j,t)=\iint q_j(x,y)u_{(0,0,0)}(x_j-x,y_j-y,t)dxdy \quad (1)$$

Then, by adopting the concept of superposition of ATMI model, the temperature of processor can be calculate by

$$T_p=\sum_{j\in N}\gamma_j u_j(x_j,y_j,t) \quad (2)$$

We give a simpler and faster core power estimation method. Generally, most of the processors include some dedicated hardware performance counters for debugging and measurement. Hardware performance counters includes event signals generated by processor functional units, event detectors detecting these signals and triggering the counters, and configured counters incrementing according to the triggers. As the read operation of performance counters can be finished by only several assembly instructions, we can read one or more performance registers to calculate each core power in every or several time slots which ensures no influence on the timing requirements of real-time systems. The performance counter used here includes cycle counts, the number of floating point register accesses, and instructions executed. A single core power density is:

$$q_j(x,y)=\sum_{v=1}^{m}a_v t_v / S_j \qquad (v=1,2,\cdots,m) \quad (3)$$

We adopt event power a_v as in [18], which is depicted in Table IV.

TABLE IV.
EVENT POWER VALUE a_v

Event	Event Power(W)
time stamp counter	11.23
unhalted cycles	14.53
retired branches	26.79

Substituting Eq.3 in Eq.2, we can get Eq.4.

$$T_p=\sum_{j\in N}\gamma_j\iint\frac{\sum_{v=1}^{m}a_v t_v}{S_j}u_{(0,0,0)}(x_j-x,y_j-y,t)dxdy \quad (4)$$

IV. SIMULATION

The GTFTES scheduling algorithm for multi-core systems is implemented in our EASS (Energy-Aware Scheduling Simulator). In this section, the experimental setup, algorithm evaluation methodologies and simulation results is presented.

A. Experimental Setup

*1) Platform:*To verify the proposed algorithm, the ATMI[17] thermal model is adopted. The ATMI thermal model takes power traces as input and outputs the steady or transient temperatures. The ATMI parameters in our experiment are listed in Table V.

TABLE V.
ATMI PARAMETERS

Parameters	Unit	Value
Heat Sink width	Meter	0.10
Heat sink thermal resistance	Meter	0.3
Copper thickness	Meter	5e-3
Silicon thickness	Meter	5e-4
Interface thickness	Meter	1e-4
Interface thermal conductivity	W/mK	4

Artificial Thread : In our simulator, we use artificial thread which is defined as a C struct (Listed as follows) for simulation.

```
struct task { /* task id */
        int tid ;
        /* thread ' s power density contribute*/
        float power_density_contribution;
        /* task' s running core */
        int execusive_core;
        /* deadline of a task*/
        int deadline ;
        /* arrivial time of a task*/
        int arrivial_time ;
        /* execution time of a task*/
        int execution_time ;

        /* worst case execution time of a task*/
        int worst_case_execution_time ;

        /* thread state
         RUNNINGSTATE: the task is running now.
        EADYSTATE: the task is ready for running
        */
        int state ;

        /*The simulation step that a processes may occupy*/
        int duration_step ;

        /*The range that a task's simulation step number*/
        int duration_step_range ;

        /*The deadline penalty that a task has to pay*/
        float deadline_penalty;

        /*The thermal penalty that a task has to pay*/
```

```
    float thermal penalty ;
};
```

Tasks are classified into Cool, Warm and Hot three types, as listed in TABLE VI. A sample task set is given in Table VI.

TABLE VI.
A SAMPLE TASK SET.

Task	Arrival Time	Deadline	Power Contribution	Execution Core	Thermal Type
τ_1	31	37	$13.131*10^5$	1	Hot
τ_2	7	37	$11.382*10^5$	1	Hot
τ_3	3	42	$14.308*10^5$	1	Hot
τ_4	4	30	$10.224*10^5$	1	Hot
τ_5	4	40	$6.538*10^5$	2	Warm
τ_6	22	40	$8.257*10^5$	2	Warm
τ_7	16	18	$0.641*10^5$	2	Cold
τ_8	10	20	$4.814*10^5$	2	Cold
τ_9	27	30	$8.126*10^5$	2	Warm
τ_{10}	9	14	$4.972*10^5$	2	Cold
τ_{11}	30	36	$11.221*10^5$	3	Hot
τ_{12}	9	35	$11.333*10^5$	3	Hot
τ_{13}	5	41	$13.314*10^5$	3	Hot
τ_{14}	3	29	$10.226*10^5$	3	Hot
τ_{15}	2	36	$7.548*10^5$	3	Warm
τ_{16}	20	39	$6.153*10^5$	4	Warm
τ_{17}	17	20	$0.289*10^5$	4	Cold
τ_{18}	11	21	$5.113*10^5$	4	Cold
τ_{19}	26	31	$8.143*10^5$	4	Warm
τ_{20}	10	13	$4.953*10^5$	4	Cold

B. Methodology

ATMI can model different parts of processors, such as Floating Point Unit (FPU), and consider their influences on the processor energy. We simulate cores of a multi-core processor as the ATMI parts of a processor. Since our algorithm focuses on the core energy consumption, we only simulate the cores without other parts of processor (FPU, Instruction Cache, Data Cache, L2 Cache and so on). Of course, these omitted parts have influences on the processor's temperature, for simplicity, we suppose that the influence is zero. We rewrite ATMI to implement our core temperature calculation method and thread thermal contribution prediction. Every core corresponds to an ATMI image. The ATMI outputs processor temperature $T_{processor}$ and $T_{core,i}$, where $1 \leqslant i \leqslant n$ is the core index, i.e., we first consider a 4-core processor.

The ultimate goal of the GTFTES is to get uniform temperature and power distribution. So, in the simulation we have to decide the factor (power or temperature) to evaluate the efficiency of the algorithm. As we all know, excessive power consumption leads to short battery life, higher utility costs, and large currents in interconnect, and elevated temperatures. Also, the power can fluctuate in a large range in a short time, while the temperature may not. All those come to our decision to use temperature as a measure of the GTFTES algorithm.

We simulate the NBS-EATA algorithms in [9], a pure Tit-for-Tat algorithm where cores bid without consideration of corporation and our GTFTES algorithm. Firstly, we simulate the temperature differences irrespective of the task types (aperiodic tasks, periodic tasks and sporadic tasks). After that, task types are involved in the simulation, and in this part, the temperature distributions are evaluated. Finally, the simulation is extended to an eight-core processor.

C. Simulation Results of Temperature Difference

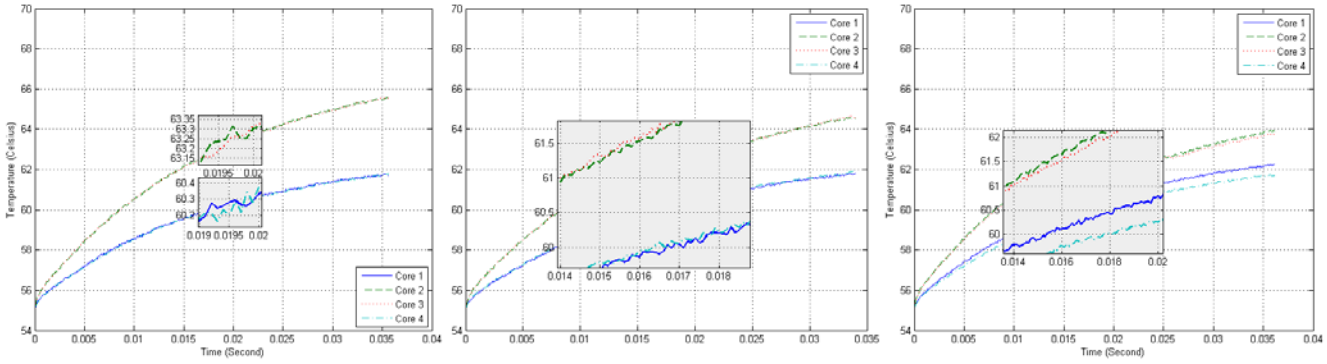
The temperature difference between cores is an important evaluation factor for the algorithms, since the smaller the difference the more uniform power density map will the algorithms provide.

We simulate different task sets with 50, 100 and 200 tasks respectively. The GTFTES algorithm performs better than NBS-EATA and a pure Tit-for-Tat algorithm. For the sake of easy reference, we call the pure Tit-for-Tat algorithm PTFT algorithm.

Our processor floorspan contains four cores sitting in a line. Thus, the two middle cores in the processor get higher temperatures than the other two cores. That is why, in our results, the four cores' temperatures fell into two categories: side cores and middle cores (As illustrated in Figure 3-5, the upper two lines are the temperature of middle cores, and the lowers are the side cores'.). For the sake of clearness, we set one or two amplifiers (grey blocks in each subfigures of Figure 3-5.) to get a close look of the data.

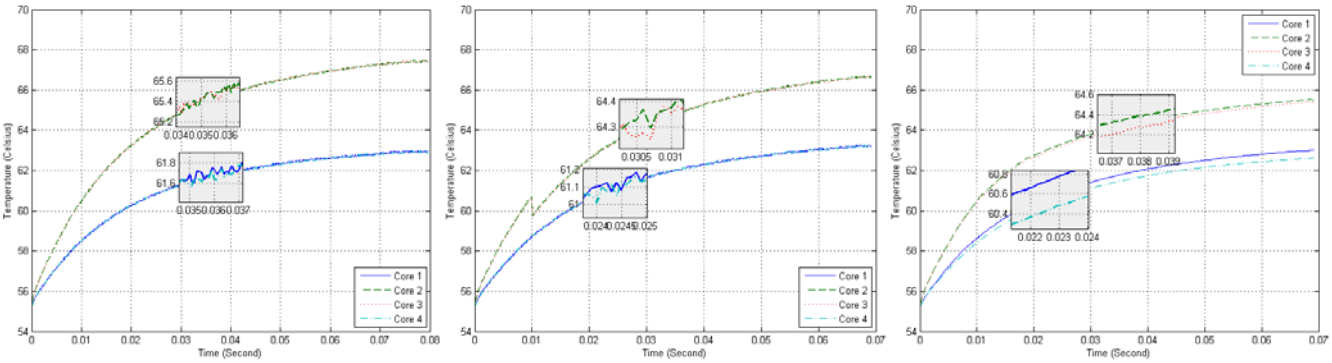
Take Figure 4 as an example, during the time period from 0 to 0.08, the temperature difference of the two groups of cores grows steadily from nearly 0 to almost 5 centigrade. In the GTFTES algorithm, the cores sometimes are forced to cooperate or they may choose to cooperate in a round of auction. It means cores that choose to cooperate may lose the auction because of its high thermal status, consequently leading to a lower temperature. Therefore, as subfigures c in Figure 3-5 show, the GTFTES algorithm finally gets a lowest temperature difference among the three algorithms.

Also, in Figure 5 which shows a temperature variation of a run of 200 tasks, GTFTES also shows a smaller difference between the two sets of core (around one centigrade lower than the NBS-EATA algorithm and two centigrade lower than the PTFT algorithm). Thus, the temperature difference between the central and side part of the processor is smaller.



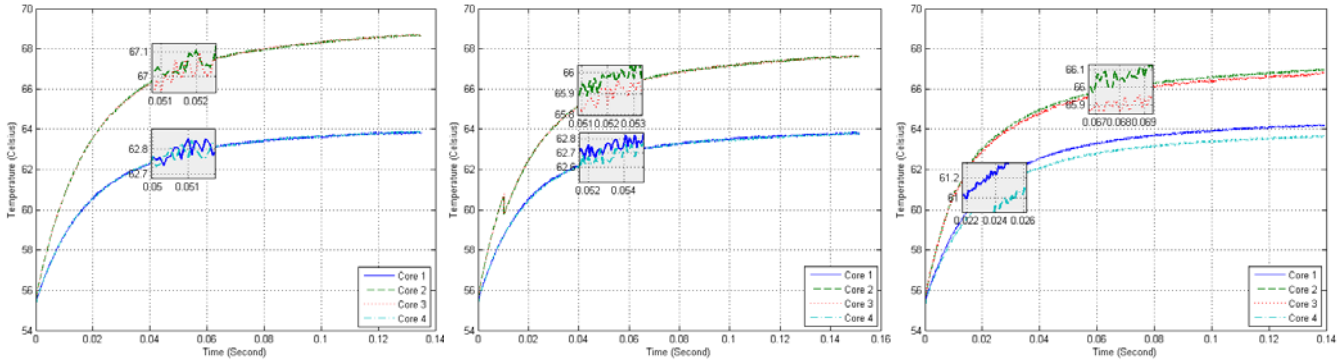
(a) PTFT algorithm (b) NBS-EATA (c) GTFTES

Figure 3. Temperature variations with 50 tasks.



(a) PTFT algorithm (b) NBS-EATA (c) GTFTES

Figure 4. Temperature variations with 100 tasks.



(a) PTFT algorithm (b) NBS-EATA (c) GTFTES

Figure 5. Temperature variations with 200 tasks.

D. Simulation Results of Different Types of Tasks

We generate three task sets with 250 aperiodic, periodic and sporadic tasks. Then, we simulate the GTFTES, NBS-EATA and PTFT algorithms separately. After the simulation we get numerous discrete time points and the temperature at those points. The temperature distribution on those discrete time points has to be listed to show the competence of the GTFTES algorithm. Instead to show a temperature curve, we use TABLE VII to list the statistical results in detail for a clearer illustration.

From the shaded grids of TABLE VII, we can see that the temperature range with majority tasks is decreasing gradually from ⩾68 ºC in PTFT to 66-68 ºC in NBS-EATA

and finally 64-66 ºC in GTFTES. Thus, we can say GTFTES can deal with aperiodic, periodic and sporadic tasks all with a better effect to lower the temperature of processors.

TABLE VII.
TEMPERATURE DISTRIBUTION OF PTFT, NBS-EATA AND GTFTES WITH 250 TASKS

	Percentages of the corresponding temperatures(%)								
	PTFT			NBS-EATA			GTFTES		
	Aperiodic	Periodic	Sproadic	Aperiodic	Periodic	Sproadic	Aperiodic	Periodic	Sproadic
⩽56ºC	4.22	4.51	3.52	3.35	4.16	6.17	4.23	4.16	5.16
56-58 ºC	8.45	7.16	9.16	12.54	13.17	12.46	11.66	13.27	13.89
58-60 ºC	11.46	12.16	12.45	15.32	15.33	14.33	14.52	15.13	16.11
62-64 ºC	13.3	14.33	14.33	17.97	16.38	16.85	18.87	18.48	20.83
64-66 ºC	14.98	13.88	15.75	21.21	19.72	23.31	30.24	29.51	28.44
66-68 ºC	19.1	18.11	19.21	29.24	30.51	26.42	20.21	18.72	15.31
⩾68 ºC	28.12	29.12	25.12	0	0	0	0	0	0

Note: Aperio: Aperiodic tasks; Perio: Periodic tasks; Sproa: Sporadic tasks. Gray grids represent the highest temperature range in a column.

TABLE VIII.
TEMPERATURE DISTRIBUTION OF PTFT, NBS-EATA AND GTFTES WITH 250 TASKS IN A EIGHT-CORE PROCESSOR

	Percentages of the corresponding temperatures(%)								
	PTFT			NBS-EATA			GTFTES		
	Aperiodic	Periodic	Sproadic	Aperiodic	Periodic	Sproadic	Aperiodic	Periodic	Sproadic
⩽56ºC	3.22	4.61	5.42	2.15	4.15	4.17	5.16	4.2	6.06
56-58 ºC	7.45	7.06	7.28	13.73	13.28	13.46	10.73	12.18	12.98
58-60 ºC	9.46	11.25	12.33	17.36	15.25	15.31	14.59	16.18	16.12
62-64 ºC	12.3	12.25	13.43	19.94	16.46	16.77	21.8	18.48	21.93
64-66 ºC	13.98	16.87	16.66	26.23	24.63	24.4	36.14	26.61	25.34
66-68 ºC	22.1	19.12	22.32	20.21	25.5	25.43	18.31	21.62	17.31
⩾68 ºC	31.12	28.11	22.1		0	0	0	0	0

Note: Aperio: Aperiodic tasks; Perio: Periodic tasks; Sproa: Sporadic tasks. Gray grids represent the highest temperature range in a column.

D. Simulation Results of Different Number of Cores

In order to show the core adaptability, we simulate the GTFTES algorithms under an eight-core processor. By listing the temperature distribution of this round of simulation (TABLE VIII), the similar conclusion that the GTFTES gets lower temperatures than PTFT and NBS-EATA algorithms can be reached.

V. RELATED WORKS

The majority of the power management research focuses on power management for the purpose of saving energy, not for balancing power consumptions and maintaining safe temperature levels [19-22]. While energy and temperature are closely related, power control mechanisms for energy and temperature are quite different. Although it has been shown that many energy-saving techniques do not work well in reducing peak temperature [23-25], there is still some change to design energy-balancing techniques in multi-core systems.

In this section, we generally look into the power management techniques from the perspective of temperature management, power management, and finally the game theory methods used in energy-related scheduling.

Firstly, from the perspective of temperature management, the authors [26] study the thread migration in temperature constrained multi-core processors. They show that the thermal benefit of thread migration depends on the number of threads, characteristics and ambient temperature. The thread migration algorithm makes thread exchange when a cold and a hot core are detected. They demonstrate that their method yields the same throughput with HRTM (Heat-and-Run Thread Migration) [6], but requires fewer migrations. Authors in [2] a temperature-aware task scheduling algorithm in microprocessor systems.

Secondly, several power management techniques have been proposed and applied in modern processors via either hardware or software mechanisms [3, 17, 18]. Hardware level DTM mechanisms, such as Dynamic Frequency Scaling(DFS) and Dynamic Voltage Scaling(DVS) as well as clock gating, are able to reduce processors temperature effectively and guarantee thermal safety, but with high performance loss. So some literature adapts to the software based mechanisms. As to the multi-core processors power management, there are some software-based mechanisms, such as an power density management approach based on operating system is studied in [6]. The proposed method has two key components: SMT thread assignment and CMP thermal migration. Within the heat-and-run the SMT thread assignment attempts to increase processor-resource utilization by co-scheduling threads using complementary resources; the CMP thread migration cools overheat by migrating threads. They show that their mechanism achieves 9% higher average throughout than stop-go and 6% higher average throughout than DVS. The authors [27] propose a

method to balance power consumption in multiprocessor systems. They use performance counters to create energy profiles which is used to describe the energy characteristics of individual tasks, and distribute energy consumption of all CPUs of a system. Their results show a 5% increase in throughput for a system with all CPUs busy by avoiding the throttling of processors. As stated above, these methods pay their attentions on the power management rather than the temperature management. The authors [28] present a method to minimize the total energy consumption and guarantee deadline of each task. They present a mixed-integer linear programming model for the NP-complete scheduling problem and solve it for moderate sized problem instances using a public-domain solver. For larger task sets, they present a novel low-energy earliest deadline first (LEDF) scheduling algorithm and apply it to two real-life task sets. However, they do not pay much attention to multi-core systems and energy balancing management.

Finally, authors in [8, 9] propose a corporative game theoretic method to address the problem of power-aware scheduling/mapping of tasks onto heterogeneous and homogeneous multi-core processor architectures. They consider the problem as a multi-objective optimization problem. However, they did not pay much attention to the temperature problems which is a key factor in the energy consumption[25].

VI. Conclusion

In this article, we propose a generalized tit-for-tat based energy-aware scheduling algorithm, namely GTFTES, for multi-core systems. Right before each auction, each core will decide to cooperate or not by the hardness factor. If it decides to cooperate, it will bid according to its power status. Otherwise, it will bid according to the average execution time of the tasks. Moreover, we have given core temperature calculation method which is used in the result analysis stage. In GTFTES, the scheduler forces the cores to cooperate if some core is thermal saturated. Simulation results show that the proposed game can surely reduce the temperature between different groups of cores and avoid the hotspot of the processor.

Acknowledgment

This paper is partially supported by the National Natural Science Foundation of China under Grants No.60703101 and No. 60903153, and the Fundamental Research Funds for the Central Universities.

References

[1] S. Borkar, "Design Challenges of Technology Scaling," *IEEE Micro,* vol. 19, pp. 23-29, 1999.

[2] M. Chrobak*, et al.*, "Algorithms for Temperature-Aware Task Scheduling in Microprocessor Systems," presented at the Proceedings of the 4th international conference on Algorithmic Aspects in Information and Management, Shanghai, China, 2008.

[3] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management: Classification and New Exploration," *SIGARCH Comput. Archit. News,* vol. 34, pp. 78-88, 2006.

[4] N. S. Fadi, "Simulation and Performance Analysis of Multi-core Thread Scheduling and Migration Algorithms," 2010, pp. 895-900.

[5] S. Heo*, et al.*, "Reducing power density through activity migration," presented at the Proceedings of the 2003 international symposium on Low power electronics and design, Seoul, Korea, 2003.

[6] M. Gomaa*, et al.*, "Heat-and-run: leveraging SMT and CMP to manage power density through the operating system," *SIGOPS Oper. Syst. Rev.,* vol. 38, pp. 260-270, 2004.

[7] J. A. Winter and D. H. Albonesi, "Scheduling algorithms for unpredictably heterogeneous CMP architectures," in *Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on*, 2008, pp. 42-51.

[8] I. Ahmad*, et al.*, "Energy-Constrained Scheduling of DAGs on Multi-core Processors," in *Contemporary Computing*. vol. 40, S. Ranka*, et al.*, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 592-603.

[9] I. Ahmad*, et al.*, "Using game theory for scheduling tasks on multi-core processors for simultaneous optimization of performance and energy," in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, 2008, pp. 1-6.

[10] R. K. Dash*, et al.*, "Market-Based Task Allocation Mechanisms for Limited-Capacity Suppliers," *Systems, Man and Cybernetics, Part A, IEEE Transactions on,* vol. 37, pp. 391-405, 2007.

[11] Y. Wang*, et al.*, "Economic-Inspired Truthful Reputation Feedback Mechanism in P2P Networks," in *FTDCS '07: Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems*, 2007, pp. 80-88.

[12] A. Pekeč and M. Rothkopf, "Combinatorial Auction Design," *Management Science,* vol. 49, pp. 1485-1503, 2003.

[13] M. H. Rothkopf*, et al.*, "Why Are Vickrey Auctions Rare?," *The Journal of Political Economy,* vol. 98, pp. 94-109, 1990.

[14] W. Guowei, "GTFTTS: A Generalized Tit-for-Tat Based Corporative Game for Temperature-Aware Task Scheduling in Multi-core Systems," 2010, pp. 81-88.

[15] W. Huang*, et al.*, "Hotspot: A compact thermal modeling method for CMOS VLSI systems," *IEEE Transactions on,* vol. 14, pp. 501-513, 2006.

[16] G. Wu and Z. Xu, "Temperature-aware task scheduling algorithm for soft real-time multi-core systems," *Journal of Systems and Software,* vol. 83, pp. 2579-2590, 2010.
[17] P. Michaud. (2009, *ATMI manual*. Available: http://www.irisa.fr/alf/
[18] G. Magklis*, et al.*, "Profile-based dynamic voltage and frequency scaling for a multiple clock domain microprocessor," *SIGARCH Comput. Archit. News,* vol. 31, pp. 14-27, 2003.
[19] H. Aydi*, et al.*, "Dynamic and Aggressive Scheduling Techniques for Power-Aware Real-Time Systems," presented at the Proceedings of the 22nd IEEE Real-Time Systems Symposium, 2001.
[20] Y. Liu and A. K. Mok, "An Integrated Approach for Applying Dynamic Voltage Scaling to Hard Real-Time Systems," presented at the Proceedings of the The 9th IEEE Real-Time and Embedded Technology and Applications Symposium, 2003.
[21] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," presented at the Proceedings of the eighteenth ACM symposium on Operating systems principles, Banff, Alberta, Canada, 2001.
[22] A. Qadi*, et al.*, "A Dynamic Voltage Scaling Algorithm for Sporadic Tasks," presented at the Proceedings of the 24th IEEE International Real-Time Systems Symposium, 2003.
[23] N. Bansal*, et al.*, "Dynamic Speed Scaling to Manage Energy and Temperature," presented at the Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004.
[24] N. Bansal*, et al.*, "Speed scaling to manage energy and temperature," *J. ACM,* vol. 54, pp. 1-39, 2007.
[25] K. Skadron*, et al.*, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.,* vol. 1, pp. 94-125, 2004.
[26] P. Michaud*, et al.*, "A study of thread migration in temperature-constrained multicores," *ACM Trans. Archit. Code Optim.,* vol. 4, p. 9, 2007.
[27] A. Merkel and F. Bellosa, "Balancing power consumption in multiprocessor systems," *SIGOPS Oper. Syst. Rev.,* vol. 40, pp. 403-414, 2006.
[28] V. Swaminathan and K. Chakrabarty, "Real-time task scheduling for energy-aware embedded systems," *Journal of the Franklin Institute,* vol. 338, pp. 729-750, 2001.

Guowei Wu, born in 1973, male, is currently an associate professor in School of Software, Dalian University of Technology. He received his Ph.D degree from Harbin Engineering University, China, in 2003. His research interests include real-time embedded system, cyber-physical systems(CPS), wireless sensor network.

Zichuan Xu, born in 1986, male, received Master degree and B.E degree (all with highest honor) from Dalian University of Technology, China. His research interests include approximation algorithms, temperature-aware scheduling, cyber-physical systems (CPS), game theory, green computing.

Qiufen Xia, born in 1987, received the M.A. from Dalian University of Technology, Dalian in 2009. She is a master student of software school, Dalian University of Technology. Her study interests include grid resources scheduling, game theory, network security and cloud computing.

Jiankang Ren, born in 1986, male, received the B.E. and M.E. degrees from Dalian University of Technology, China, in 2008 and 2011, respectively. His current research interests involve fault-tolerant and reliable computation in cloud computing, QoS and managing strategies, security and privacy in Body Sensor Networks and Cyber Physical Systems.

A Failure Self-recovery Strategy with Balanced Energy Consumption for Wireless Ad Hoc Networks

Tie Qiu, Wei Wang, Feng Xia*, Guowei Wu, and Yu Zhou
School of Software, Dalian University of Technology, Dalian 116620, China
Email: qiutie@dlut.edu.cn; f.xia@ieee.org

Abstract—**In energy constrained wireless sensor networks, it is significant to make full use of the limited energy and maximize the network lifetime even when facing some unexpected situation. In this paper, all sensor nodes are grouped into clusters, and for each cluster, it has a mobile cluster head to manage the whole cluster. We consider an emergent situation that one of the mobile cluster heads is broken down, and hence the whole cluster is consequently out of work. An efficient approach is proposed for recovering the failure cluster by selecting multiple static sensor nodes as the cluster heads to collect packets and transmit them to the sink node. Improved simulated annealing algorithm is utilized to achieve the uniform deployment of the cluster heads. The new cluster heads are dynamically changed in order to keep balanced energy consumption. Among the new cluster heads, packets are transmitted through multi-hop forwarding path which is cost-lowest path found by Dijkstra's algorithm. A balanced energy consumption model is provided to help find the cost-lowest path and prolong the lifetime of the network. The forwarding path is updated dynamically according to the cost of the path and residual energy of the node in that path. The experimental results show that the failure cluster is recovered and the lifetime of the cluster is prolonged.**

Index Terms— **wireless sensor networks, mobile ad-hoc networks, energy consumption, failure recovery**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are widely used in environmental monitoring, disaster relief, health care and so on. WSNs are composed of large numbers of sensor nodes which are battery-powered, with sensing and limited computation as well as communication capabilities. Their major work is to sense the environment and route data packets to the Base Station (BS) via multi-hop path [1, 2]. Due to the battery constraint, the sensor nodes collect the useful information and transmit them over a long time period [3]. And the majority of energy consumption is expended on forwarding the packets [4]. How to provide a balanced energy consumption achieving maximum extension of lifetime and improving Quality of Service (QoS) of network has become a research focus in recent years.

Large numbers of sensor nodes are grouped into clusters [5], and each cluster has a cluster head. Vupputuri et al. [1] propose an efficient way using a mobile data collector(DC) as the cluster head to collect the data from static sensor nodes. Then the DCs aggregate the data and transmit it to the BS. DCs have the capability of motion which can be controlled. Because the energy of the sensor nodes is mainly expended on transmitting the data, hence in a multi-hop network, it will consume more energy if the node is closer to the sink node [6]. Consequently, DCs should change their location dynamically according to the energy of different area. DCs will move to the area which is of high energy. DCs play an important role in the WSNs, however, there is a shortcoming that if one of the DCs is out of work, whole cluster won't work anymore. So it is necessary to focus on self-reorganization for the WSNs and balanced energy consumption in unexpected situation.

In this paper, we propose a solution of self-recovery strategy with balanced energy consumption in wireless ad hoc networks with the failure cluster which is out of work. Simultaneously, the energy consumption of the node is minimized and we can keep the original performance of the whole WSNs. When meeting the unexpected situation, the WSNs can recover the cluster by itself. The rest of the paper is organized as follows. Section II describes the related work and the problem statement. In Section III, we define the energy model, and propose an improved simulated annealing algorithm to achieve the uniform deployment, besides we use the Dijkstra's algorithm to find the cost-lowest forwarding path. In Section IV, we will discuss the simulation results. Finally, Section V concludes the paper.

II. RELATED WORK AND PROBLEM STATEMENT

A. Related Work

Due to the strict energy constraints of the nodes, it is extremely essential to optimize the energy consumption for WSNs. Energy consumption consists of transmission cost

*Corresponding author

doi:10.4304/jcp.7.1.116-123

and aggregation cost [7]. And there is also another energy cost for transmission and reception of the packets and an energy cost depending on the distance between two nodes in transmission to balance the energy consumption [8]. In addition, the node can be set to several states, such as low-energy consumption state, transmission state, receiving state and so on. In LEICP [9], in order to prolong the lifetime of the WSNs, a fitness function is defined to balance the energy consumption in every cluster according to the residual energy and positions of nodes. In [10], the authors present a comprehensive energy model for a fully function wireless sensor network, and the model is divided into 2 parts, energy consumption due to synchronization and the energy consumption due to data transmission. Besides, the energy for empty frames and missed part are also analyzed. In this paper, we also provide an energy consumption model like [10]. However, it is important to apply the energy model in finding the cost-lowest path and the formation of the cluster. Moreover, in ad-hoc WSNs, the forwarding path should vary dynamically.

In order to optimize energy consumption and maximize the life time of the WSNs, balanced sensor deployment as well as cost-lowest path is found by Dijkstra's algorithm in [11] and [12]. Dijkstra's algorithm will find the cost-lowest path based on the path distance, while we propose to apply the energy model to the Dijkstra's algorithm to select the optimized path. So it is reasonable to have uniform energy consumption.

It is of great significance for the cluster heads to have a better coverage of all the sensor nodes in its cluster [13]. Recently, there are many ways to detect the sensor and maximize the coverage. In [6], the authors propose that cluster heads perform parallel particle swarm optimization to maximize the coverage matrix. In LEACH, they use simulated annealing to find the most optimized location for the cluster heads. In [14], simulated annealing is also utilized to optimize localization. In the procedure of the simulated annealing algorithm, it will select the adjacent nodes to be compared with the current selected cluster heads, and the group of nodes with the minimum distance sum which is calculated from all the other nodes to the cluster heads will be the cluster heads. Consequently, it will make a contribution to the uniform distribution of the nodes. In [15], the parallel distributed self-organization clustering protocol based on clustering architecture is proposed. And the WSNs are partitioned into many small logic zones distributed uniformly according to the geography locations of the nodes. In this paper, we select cluster head not only based on the distance, but also the number of the times for which the node has been the cluster head, because the cluster head will be updated dynamically. In [16], the method is to dynamically schedule sensors' work cycles or sleep cycles in the heterogeneous WSNs. The author used a multiple criteria decision making method to optimize the sleep scheduling process. These studies have successfully obtained good results in energy optimization and extension of the maximum lifetime of WSNs, but failure self-recovery strategy with balanced energy consumption for WSNs with multiple clusters is rarely mentioned.

B. Problem Statement

However, almost all recent papers assume that the nodes are all in well condition. But we cannot deny that majority of the sensor nodes are exposed in the nature, perhaps it will break down for physical reasons (e.g. components of the circuit is damaged by high temperature or water) or technical reasons (e.g. a software bug in the system), especially if the cluster head is out of work, the performance of the whole WSNs will go down. So it is important and necessary to detect and solve this problem quickly.

A WSN is divided into multiple clusters, every cluster has one Mobile Cluster Head (MCH), which will collect the packets from sensor nodes in their own cluster, and then send the packets to BS. If a MCH is out of work, packets of the whole cluster are missed. As showed in Figure 1, the red symbol is the broken down MCH. We suggest that multiple sensor nodes should be selected in that cluster, which will act as the role of the failure MCH. The new cluster heads will take responsibility of collecting the packets from member sensor nodes. Besides, among the new cluster heads, a cost-lowest forwarding path will be found to transmit the packets to the adjacent MCH or BS. So the balance of the energy consumption in the failure cluster needs to be focused on to maximize the lifetime of the whole cluster in this unexpected situation.

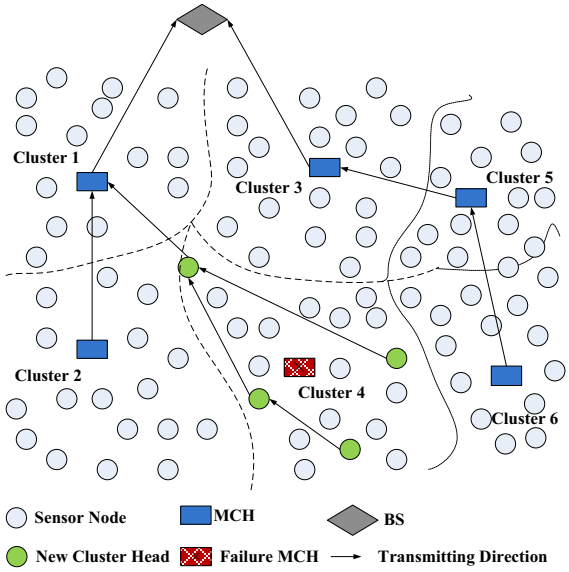


Figure 1. A WSN with failure MCH

We simply describe a WSN with failure MCH and present the approach as showed in Figure 1. The WSN is divided into six clusters (cluster 1-6). Data packets of sensor nodes are collected by Cluster head for each cluster in the WSN. If the cluster head close to the base station, the data packets will be collected directly to the base station. If the

cluster head far away from the base station, data packets are sent to the base station through other cluster heads. As showed in Figure 1, MCH of cluster 4 breaks down suddenly, and the whole cluster is consequently out of work. If a sensor node is selected to substitute the breakdown MCH, the node energy will be consumed rapidly, which leads to the extremely early death of the node. This paper is aimed at re-organizing the failure cluster by balanced energy consumption of the new cluster head. The new cluster heads and the forwarding path found by Dijkstra's algorithm are constantly updated, ultimately achieving balanced and minimum energy consumption to prolong the network lifetime.

III. MODELING AND ANALYSIS

We propose a strategy to re-organize the sensors by themselves. Our solution is to filter out the eligible sensor nodes to be cluster heads at first. The second step is to find the cost-lowest path according to Dijkstra's algorithm which is based on the energy consumption model. Finally, the path weight will be changed dynamically based on the energy model. Consequently, the forwarding path is also changing during the packets transmission. Then the whole networks can work normally. Furthermore, our contribution is to recover breakdown cluster without extra mobile cluster head. More importantly, the life time is almost the same as the original networks.

A. Eligibility of Sensor Nodes

First of all, the eligible sensor nodes need to be filtered out. In LEACH, it selects the eligible sensor nodes simply according to the energy, while the selection process is based on the location and distance of the nodes. But this procedure may cause that the node with high energy as well as good location (specifically, it may be relatively close to the centre of the area) may be selected as the cluster head frequently, which will have a negative effect on the energy balance in the whole cluster. As a result, we select the cluster heads according to their residual energy and the times for which they have been the cluster head. For this restriction, the sensor is eligible, only if the energy of the sensor surpasses the average energy of the whole cluster, and the number of times it has been the cluster head is less than the average times. Thus not all the sensor nodes which surpass the average energy are equal. The nodes are more likely to be selected which have never been the cluster head. It is defined as:

If $E_i > \overline{E}$ and $P_i > \overline{P}$ then the *i-th* node is eligible.

$$\overline{E} = \frac{\sum_{i=0}^{nn} E_i}{nn} \tag{1}$$

$$\overline{P} = \frac{\sum_{i=0}^{nn} P_i}{nn} \tag{2}$$

where *nn* is the number of the sensor nodes in this cluster, E_i is the residual energy of *i-th* sensor node's, $\overline{E}$ is the average energy of the whole sensors in the cluster, P_i is the number of the times for which *i-th* node has been the cluster head, $\overline{P}$ is average times of the whole sensors. The nodes' energy and the eligible of the sensor nodes vary dynamically. In every round when changing the cluster heads, eligibility needs to be calculated again.

B. Uniform Deployment for New Cluster Heads

The uniform distribution of the cluster heads is also important to the energy balance of the nodes. We discuss how to select the cluster heads from eligible nodes to make the cluster heads' location well-distributed based on the distance and energy. Thus coverage and the energy efficiency can be provided for WSNs.

We have improved the model of simulated annealing, and we add our evaluating standard into the annealing model to optimize the process. The theory of simulating annealing originated in the physical theory, and the concept is based on the manner in which liquids freeze or metals re-crystallize in the process of annealing [14], which can be used to locate a good approximation to the global optimum of a given function in a large search area. Here we can use this theory to select a location of the cluster heads to perform the best coverage and the energy efficiency. The process is:

When the algorithm of simulated annealing starts, the sensor nodes are initialized to be a high energy level. Then multiple sensor nodes are randomly selected from the eligible sensor nodes as the initial cluster heads. The number of the cluster heads depends on the scale of the cluster, packets length, and so on [16]. Next, the optimization process will begin. At first, the cost of the initial cluster heads will be calculated. Then each cluster head's neighbor will be randomly selected as a new group of cluster heads. Neighbors' existence must be confirmed. And the cost of the new cluster head is calculated, if the cost of the new cluster heads is less than the current cluster heads, the group of new cluster heads will substitute current cluster heads in this step, otherwise current cluster head is the same. At last, previous process is repeated until the iteration is at the end. Thus the best choice of the new cluster heads is found out.

The cost is defined according to not only the distance of sensor nodes but the energy state of each one.

$$\text{cost} = \sum_{j=1}^{nn} \alpha \times MIN(D_{ij}) \times (1 + P_i / P_s + \beta / E_i)(i = 1, 2, 3 \ldots N_c) \tag{3}$$

where α , β are const numbers which are suitable to different specific WSNs. D_{ij} is the distance between *i-th* sensor node and *j-th* sensor, the function *MIN* will find the minimum D_{ij} from other cluster heads to the *j-th* sensor node. *i* is the sequence number of cluster heads which is from 1 to N_c. N_c is the number of the cluster heads. *nn* is the number of the sensor nodes, P_j is the number of the times for which the *j-th* cluster head has been the cluster head, P_s

is the total number of the times, E_i is the residual energy of *i-th* cluster head. With this energy cost model used in the simulated annealing, we can obtain a better location of the cluster heads.

Algorithm 1: Find Optimized Cluster Heads

Begin

Step 1. Initialize D_{ij}, P_i, P_s, E_i, which is defined in equation (3);

Step 2. Randomly select N_c cluster heads from eligible sensor nodes. The state S is initialized, $S = \{1,2,.. N_c\}$;

Step 3. Calculate cost of current cluster heads *CC* by equation (3);

Step 4. Set minimum cost, *MIN_C = CC;*

Step 5. Set iteration condition *ITER*(It is set to be 1000 in this paper), and set the iteration value *K*=0;

Step 6. Find the cluster heads with lowest cost:

Step 6.1 Randomly select neighbor sensor nodes of *S* as new cluster heads;

Step 6.2 Calculate the cost of new cluster heads *CN*;

Step 6.3 Decide whether to change current cluster head to be new cluster heads

Step 6.3.1 According to LEACH , set $ck = 1000 \times \exp(-K / 20)$;

Step 6.3.2 Calculate probability P_K of changing the current cluster heads:

If (*CN*<*MIN_C*) $P_K = 1$

else if (*CN*==*MIN_C*) $P_K = 0$

else $P_K = \exp(-(CN - MIN\_C) / ck)$

Step 6.3.3 let *rand* = a random real number in (0, 1);

Step 6.3.4 if (*rand*< P_K)

Change *S* and current cluster heads with new cluster heads

MIN_C = CN

Step 6.4 *K*=*K*+1, if (*K*<*ITER*) go to **Step 6.1**, else iteration is finished and return current cluster heads;

End

In Algorithm 1, *CN* is the energy cost of new cluster heads found from the neighbor. *CC* is the energy cost using current cluster heads. *MIN_C* is recently the minimum cost. *ITER* is the loop times in which we predict that the best location of the cluster heads will be found. *S* is the state, namely, the neighbor nodes will be picked around *S*, and it is initialized to be current cluster heads. The algorithm presents that, If *CN* is less than *MIN_C*, new cluster heads becomes new optimum and the state *S* will be changed to new cluster heads. Otherwise, new cluster heads may still become new optimum with a non-zero probability set below. Because, if *CN* is less than *MIN_C*, P_K will be set to 1, so the random number distributed between 0 and 1 must be less than P_K, then the current cluster heads must be replaced by new cluster heads. However, if *CN* surpasses *MIN_C*, the probability of changing state *S* is not zero, but an extremely small number.

This algorithm has a relatively high complexity. However, it is the BS's responsibility to run this algorithm to reduce the extra consumption of sensor nodes. The location information of all sensor nodes needed in the algorithm is stored in the BS in advance. It will not vary due to the static sensor nodes. The energy level of each node and the times for which sensor node has been the cluster head will be attached in the data packet to be sent to the BS. Then after the process of the BS, BS needs to return the selection result to the selected cluster heads.

Now the new cluster heads are set up. Besides, by this algorithm, each sensor nodes have acknowledged that which cluster head it will be managed by, according to the minimum distance between itself and the cluster head. Then the sensor node will begin to transmit packet to its cluster head. Each packet from the cluster head will be attached with the energy value of themselves. And all of this will be send to the BS. BS is responsible to maintain the energy of the whole breakdown cluster. Then, once the BS finds any one of the cluster heads' energy is below the threshold, it will re-select the cluster heads for the cluster using Algorithm 1. We define the threshold *Th* as:

$$Th = \alpha \cdot \overline{E}^2 \tag{4}$$

In (4), α is a constant number, $\overline{E}$ is the average energy of the whole cluster. The value of α can be changed to select the occasion of changing the cluster heads to keep a balance between the uniform energy distribution of all sensor nodes and the times of changing the cluster head, which will finally lead to a best performance.

C. Energy Consumption Model

There must be immense and wasteful energy consumption if all the cluster heads in the failure cluster send their packets directly to the BS or adjacent MCH which is perhaps far away. It is effective to set up a multi-hop topology for the cluster heads to save energy. We suggest each cluster head select a cost-lowest way from itself to BS or adjacent MCH through multiple other cluster heads. Dijkstra's algorithm is utilized to select the cost-lowest way, and the way weight is defined as the syntheses of distance and energy.

We divide the energy consumption into two parts for each node. One is energy consumption of transmission, the other is reception consumption. They are defined separately. The reception energy consumption is a const value *C*, and the transmission part is obtained by equation (5):

$$E_t = \alpha \cdot E + \beta \cdot b \tag{5}$$

$$E = \lambda L^2 = \lambda \cdot ((ch\_x_i - ch\_x_j)^2 + (ch\_y_i \cdot ch\_y_j)^2) \cdot b \tag{6}$$

$$E_r = C \tag{7}$$

where E_t is the transmission consumption, E is the cost of radio based on the distance between two cluster heads, b is data size of the packet. If b is a constant number, E_t will be just related to the radio energy. *ch_x* and *ch_y* are the coordinate of the cluster head. E_r is reception energy.

D. Cost-lowest Path by Dijkstra's Algorithm

As what is presented previously, among the cluster heads, it is expensive if every cluster head transmit the packets directly to the BS or the adjacent MCH. It is necessary to find the cost-lowest way to the sink node. The forwarding path is probably a one-hop directly, also perhaps a multi-hop path, which all depends on the cost of the path.

In [7], Dijkstra's algorithm has been utilized to find the cost-lowest path. However, they just take the distance between two nodes into consideration. Considering that one of the cluster heads' location is so well that all the other cluster head will select it as a vertex in the path, consequently, this cluster head will expend more energy than the other cluster heads. It will cause an unbalanced energy deployment in the cluster heads, which finally results in that this group of cluster heads need to be changed so early according to equation (4). In order to address this problem, we propose a dynamical edge weight, which varies in real time. During the transmission, the path will be changed if needed according to the real-time edge weight.

Then the edge weight value between two cluster heads is defined as:

$$EW_{i,j} = E_t \cdot (1 + \theta / E_i) \quad (i \neq j) \tag{8}$$

where E_t is the transmission consumption defined in equation (5). θ is a const number. E_i is the *i-th* cluster head's residual energy. *i, j* is the number of the cluster head from *1* to N_c (defined in equation (3)). Base on this edge weight model, the weight depends on E_t and residual energy of the cluster head. Once the path is found, the E_t is a relatively stable value, and will only be changed a little if the packet size is changed, while residual energy is changing with the procedure. As the energy goes down, the edge weight will rise up correspondingly. If one of the cluster head' energy goes extremely down, the network can sense this situation, and adjust its edge weight to a high level. Then the Dijkstra's algorithm will be utilized again to find a new path for them. Obviously, the cost-lowest way is dynamically changing to achieve the best energy saving and energy balance.

Given the network Graph <*V,E*> with edge weight, Dijkstra's algorithm can find the forwarding path with minimum cost from every node to the sink. At our present situation, V is a set of the cluster heads. E is a set of edges from any one cluster head to other cluster heads. For every edge, we have an edge weight which is calculated in equation (8) stored in a matrix.

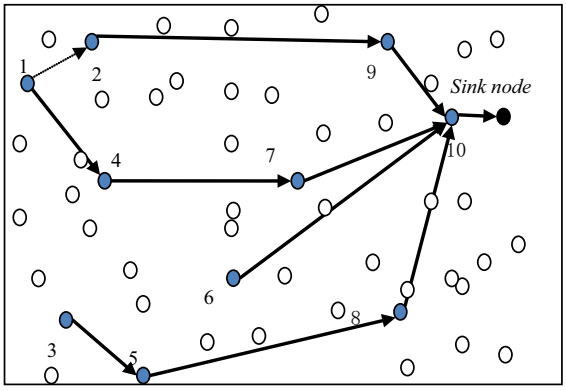


Figure 2. Cost-lowest forwarding path by Dijkstra's algorithm

At beginning, every forwarding path from the cluster head to the sink node (BS, or adjacent mobile sink node) is initialized to infinite. *S* is a set of sensor nodes whose final cost-lowest-path weights from the source cluster head have already been determined. *S* is empty initially. In the algorithm, it will repeatedly select the vertex *u* from V-S with the lowest-cost path estimated, then add it to *S*, and update the entire determined path leaving *u*. Through N_c times loop, the forwarding path from every cluster head to the sink will be set up. The running time of this Dijkstra's algorithm depends on how the min-priority queue is implemented, and we achieve the running time of $O(V^2/lgV)$. This is practical to implement the min-priority queue with a binary min heap. Furthermore, for a definite time, the edge weight matrix will be updated with the energy going down, and the Dijkstra's algorithm will be called again to re-build the path.

TABLE I. INFORMATION OF NODES

Node No.	*ch_x*	*ch_y*	E_i
1	9.6532	90.9823	0.9263
2	17.7690	102.5647	0.9256
3	26,1295	26.6744	0.9230
4	31.2210	68.2318	0.9584
5	40.6683	5.9087	0.9906
6	59.9032	35.6784	0.9756
7	75.3462	67.9984	0.9920
8	90.7640	8.0864	0.9716
9	99.9736	103.9833	0.9667
10	117.9065	82.9472	0.9575

Figure 2 shows one scenario of the forwarding path selecting. There are ten cluster heads which are numbered from 1 to 10, and one sink node. In Table 1, it shows the coordinates and energy level of each node. In order to find the cost-lowest path, according to the equation (8) and Algorithm 2, there are 4 paths selected which are cost-lowest and leads to a uniform deployment of energy. Our strength is not only focusing on the energy consumption but also energy deployment, which will make a contribution to prolonging the whole networks lifetime.

Algorithm 2: Find Cost-lowest Path

Begin
Step 1 Initialize *SN* and *V*
SN ← source nodes, such as BS or adjacent MCH
V ← cluster heads
Initialize E_t and E_i
Initialize *Np* which is used to record last node added in *SN*. Initialize *Np* = source node number.
Step 2 Update edge weight value *EW* according to equation(8)
Step 3 Iteration to find cost-lowest path
While ($V \neq \phi$)
Step 3.1 Find next node
u ← find a node number from *V* which has a minimum edge weight to *Np* from *EW*
Delete *u* from *V*.
Add *u* to set *SN*, $SN \leftarrow SN \cup u$.
Let *p* = *u*
Step 3.2 Update former cost
For each cluster head which is adjacent *u*, relax cost of the cluster-head
Update the cost-lowest path.
End While
Step 4 Return the cost-lowest path
End

In Algorithm 2, step 1 is initialization, to set *V* and *SN*, and update the edge weight value *EW*. Step 2 is iteration to extend the forwarding path to find the cost-lowest path. In step 3, every loop we select a node from *V* which has a minimum edge weight to *p* from *EW* as node *u*, then remove it from *V*, add it into *SN*. It is calculated that with the addition of *u* whether some existent paths need to be changed, namely whether the cost is lower through node *u*. If the cost is lower through *u*, the existent paths need to be updated. This procedure is repeated until all cluster heads are added in *SN*, and *V* is empty. Thus the cost-lowest path from source to the sink node is found.

IV. Numerical Calculation and Experimental Results

In this section, we will conduct some simulation experiments to analyze the energy efficiency, energy balance and lifetime time of the re-organized cluster as well as performance. In this simulation, we focus on breakdown cluster which is managed by the failure MCH. According to our strategy, static cluster head will be selected to take responsibility to transmit the packets. During the simulation, energy of each node will be recorded to prove the uniform deployment of the energy, while total mounts of packets are calculated to test lifetime of networks.

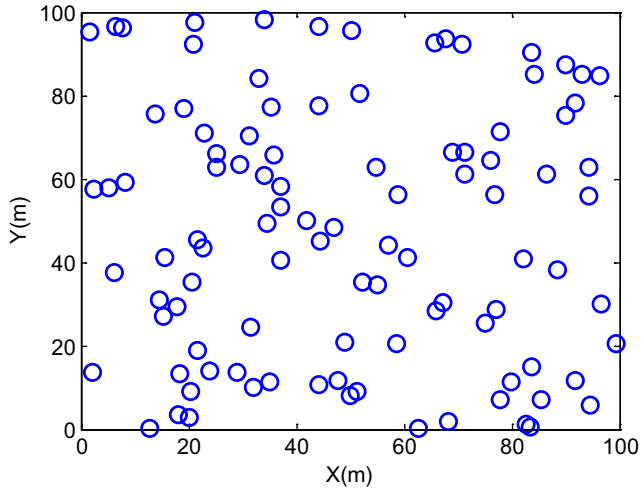


Figure 3. Sensor nodes distibution for experiment

A. Simulation Environment

Generally, we only take the breakdown cluster into consideration. It is a topology of 100 sensor nodes (i.e. *nn*=100) and one BS which is the sink unit. We assume that all these sensor nodes are distributed in an area of $100m \times 100m$, as showed in Figure 3. Distribution of the sensor nodes are generated by NS2. We suppose the initial energy of the sensor node is 1. We set the parameter $\alpha = 1 \times 10^{-7}$, $\beta = 1$, $N_c = 10$ in (3), namely, we will select 10 cluster heads as the cluster heads. In (4) we set α=0.9. In (5), $\alpha = 1 \times 10^{-7}$, $\beta = 1 \times 10^{-8}$, and we set b to a const value 1K, which means that all the packets are 1K. In (6), we set λ=1. In (7), the const reception consumption $C = 1 \times 10^{-5}$. $\theta = 1$ in (8). When the simulation ends, the total number of packets transmitted is 15263, and the total round of changing the cluster heads is 1030.

B. Balanced Eenergy Consumption

We capture 4 moments uniformly by time in the simulation to show the residual energy of all the sensor nodes, we can observe that, at any moments, the energy of the node is at the proximity level.

Figure 4 shows the energy relationship of 100 nodes at different time points. We can see that the energy difference of any two nodes' energy is less than 20% of the whole energy. If one node's energy is at the top of all nodes' energy, it wont be the top in next period, which shows that the node of high energy is more likely to be the cluster head. And the energy consumption of each sensor node is generally similar from the beginning to the end, and the energy line is going down uniformly. It can prove that in our algorithm, the energy consumption of the nodes in the failure cluster can keep balanced.

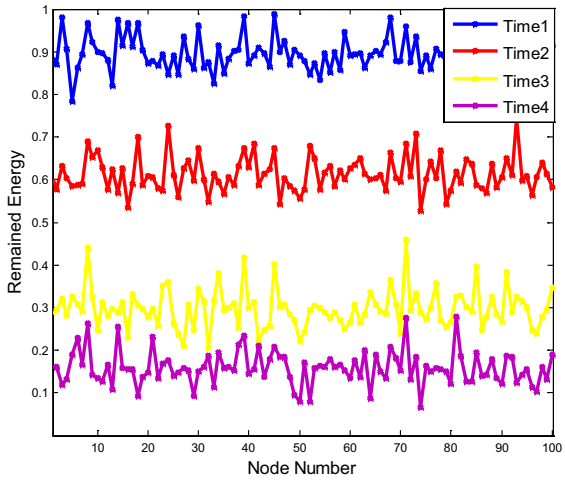


Figure 4. Remained energy of sensor nodes

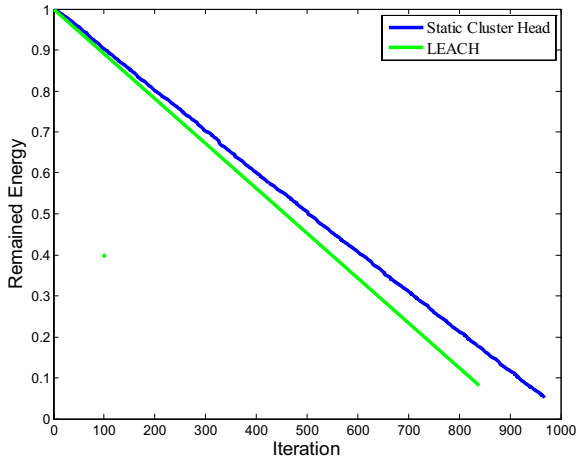


Figure 5. Average remained energy of 100 nodes

In Figure 5, the average energy of the 100 nodes is calculated in every round. It also shows that in our strategy, the average energy goes down stably, and the energy of the whole WSNs is uniformly distributed in the 100 nodes. While the lower line indicates the energy level in LEACH, the consumption of sensor nodes is larger than that in our method. The energy consumption is not uniformed due to its random selection of the member nodes. Meanwhile, the energy level of each node will fluctuate in LEACH. As for our method, using extended simulated algorithm and Dijkstra's algorithm, the consumption of the sensor nodes is uniform and minimized. It obviously reflects that lifetime of the WSNs is prolonged longer than LEACH.

Figure 6 depicts the relationship of the living nodes of the cluster. We can see that using multiple static cluster heads, 100% of the nodes are alive, which covers 90% of lifetime. However, with the MCH, at 200-th round, the nodes begin to die. But the WSNs can last a long time with MCH, because the death rate of sensor nodes is going down. Compared with the method only using DCs in [1], our method saves more energy in the whole process. Until up to the 1000th round, the nodes begin to die. In contrast, the method using only DCs starts at about the 200th round. However, in terms of the whole lifetime in our method, it is a little shorter but very close to the method using DCs.

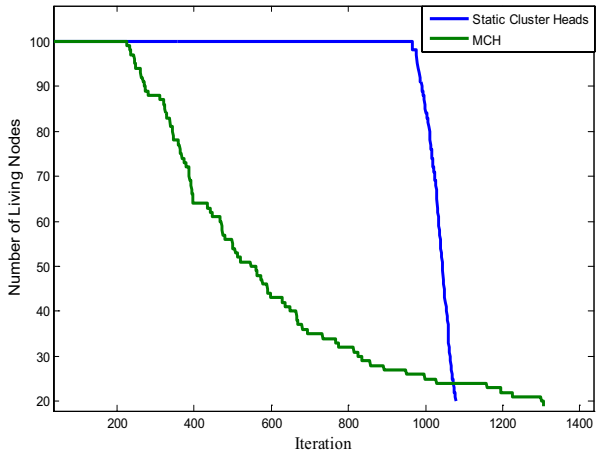


Figure 6. Number of living nodes

Figure 7 depicts the energy comparation between two situations. One situation is that cluster is recovered by multiple static cluster heads when MCH is broken down. The other is that the cluster is recovered by multiple static cluster heads at first, and then after 400 rounds, alternate MCH (AMCH) is dispatched to recover the cluster. The energy consumptions in two situations are close to each other. Due to the AMCH, the rate of the energy declines. The lifetime of WSNs is thus prolonged.

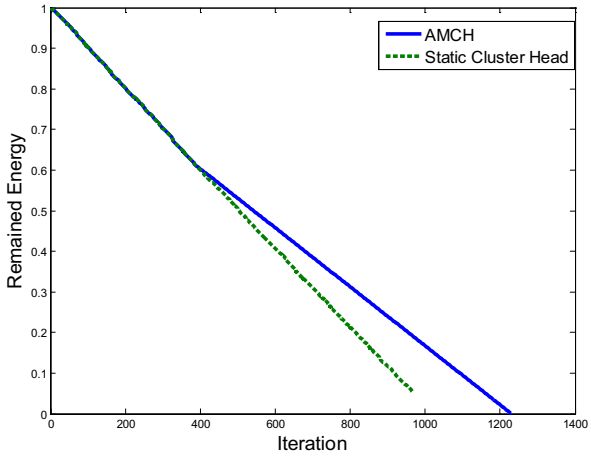


Figure 7. Remained energy

From the simulation, the contribution of our strategy is to make all the sensor nodes involved in the recovery of the cluster. All nodes have the possibility to be the cluster heads. This kind of uniform energy consumption will lead to longer lifetime. Compared with the MCH or AMCH, the performance of static cluster head is almost the same, especially in the comparison of living nodes (as shown in Figure 6) with original MCH. By using static cluster head, 100% living nodes will last longer.

V. CONCLUSIONS

The heterogeneous WSNs consist of static sensor node, mobile cluster heads and the base station. We discuss a scenario that one of the mobile cluster heads breaks down. The sensor nodes in the failure cluster will recover themselves by selecting multiple temporary cluster heads to act as role of mobile cluster head. Improved simulated annealing algorithm is utilized to achieve the uniform deployment. An energy model is applied to keep the balanced energy consumption. We use extended Dijkstra's algorithm to find the cost-lowest path based on our energy model. Our proposed solutions achieve that the failure cluster is recovered by the multiple new static cluster heads, which can work normally. Due to the balanced energy consumption, the lifetime of the WSNs is prolonged as showed in the simulation. The strength of the solution in this paper is that the breakdown networks are recovered in a short time using the static sensor nodes, while we can also keep the well performance of the network. So the whole WSNs can live longer in complex and hostile environment. Each node can participate in the management of WSNs, so the WSNs will be stronger.

This paper discusses situation that only a small number of MCHs can not work in WSNs. If there are multiple adjacent clusters do not work, our method can also self-recover, but the efficiency will be affected. Our future work will be focused on the situation that multiple adjacent MCHs do not work normally. We will re-divide clusters to increase efficiency of the node transmission in WSNs.

ACKNOWLEDGMENT

This work is partially supported by Natural Science Foundation of China under Grant No. 60903153, the Fundamental Research Funds for the Central Universities (DUT10ZD110), and the SRF for ROCS, SEM.

REFERENCES

[1] S. Vupputuri, K. K. Rahuri and C. S. R. Murthy, "Using mobile data collectors to improve network lifetime of wireless sensor networks with reliability constraints," Journal of Parallel and Distributed Computing, vol. 70, n. 7, pp. 767-778, 2010.

[2] E. Onur, C. Ersoy, H. Deliç and L. Akarun, "Surveillance with wireless sensor networks in obstruction: Breach paths as watershed contours," Computer Networks, vol. 54, n. 3, pp. 428-441, 2010.

[3] M. E. Pellenz, R. D. Souza and M. S. P. Fonseca, "Error control coding in wireless sensor networks: Trade-off between transmission and processing energy consumption," Telecommunication Systems, Special Issue: Wireless Ad Hoc, Sensor and Mesh Networks, vol. 44, n. 1-2, pp. 61-68, 2010.

[4] J. Hong, I. Jang, H. Lee, S. Yang and H. Yoon, "MRMAC: Medium reservation MAC protocol for reducing end-to-end delay and energy consumption in wireless sensor networks," IEEE Communications Letters, vol. 14, n. 7, pp. 614-616, 2010.

[5] R. W. N. Pazzi and A. Boukerche, "Mobile data collector strategy for delay-sensitive applications over wireless sensor networks," Computer Communications, vol. 31, n. 5, pp. 1028-1039, 2008.

[6] J. Sheu, P. K. Sahoo, C. Su and W. Hu, "Efficient path planning and data gathering protocols for the wireless sensor network," Computer Communications, vol. 33, n. 3, pp. 398–408, 2010.

[7] K.Yang, Y. Wu and H. Zhou. "Research of Optimal Energy Consumption Model in Wireless Sensor Network," 2010 International Conference on Computer Engineering and Technology, Proceedings, pp. 421-424, Chengdu, April, 2010.

[8] J. Jia, Z. He, J. Kuang and Y. Mu, "An Energy Consumption Balanced Clustering Algorithm for Wireless Sensor Network," The 6th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1-4, Chengdu, September, 2010.

[9] Q. Li, L. Cui, B. Zhang and Z. Fan, "A Low Energy Intelligent Clustering Protocol for Wireless Sensor Networks," IEEE International Conference on Industrial Technology, pp. 1675-1682, Chile, March, 2010.

[10] W. Dargie, X. Chao and M. K. Denko, "Modelling the energy cost of a fully operational wireless sensor network," Telecommunication Systems, Special Issue: Wireless Ad Hoc, Sensor and Mesh Networks, vol. 44, n. 1-2, pp. 3-15, 2010.

[11] X. Wang, J. Ma, S. Wang and D. Bi, "Distributed Energy Optimization for Target Tracking in Wireless Sensor Networks," IEEE Tansactions on Mobile Computing, vol. 9, n. 1, pp. 73-86, 2009.

[12] J. Tong, D. Qian, Z. Du and M. Kalisan, "Energy-Efficient Coded Routing with Selective Transmission Power for Wireless Sensor Networks," 2010 IEEE 72nd: Vehicular Technology Conference Fall, pp. 1-5, Ottawa, September, 2010.

[13] W. Wang, T. Qiu, L. Wang, F. Xia and G. Wu, "A Balanced Energy Consumption Solution for Failure Ad Hoc Wireless Sensor," 2011 IEEE International Conference on Computer Science and Automation Engineering, Shanghai, June 2011.

[14] S. K. Shekofteh, M. B. Khalkhali, M. H. Yaghmaee and H. Deldari, "Localization in Wireless Sensor Networks Using Tabu Search and Simulated Annealing," The 2nd International Conference on Computer and Automation Engineering, pp. 752-757, Singapore, February 2010.

[15] B. Yan, X. Zhou, H. Wang, F. Lang and B. Li, "An Energy-efficient Wireless Sensor Network Based on Geographical Information," Acta Automatica Sinica, vol. 34, n. 7, pp. 743-751, 2008.

[16] S. Paul, S. Nandi and I. Singh, "A Dynamic Balanced-energy Sleep Scheduling Scheme in Heterogeneous Wireless Sensor Network," The 16th International Conference on Networks, pp. 1-6, New delhi, December 2008.

An Attractive Force Model for Weighting Links in Query-Dependant Web Page Ranking

Xinyue Liu[1,2]
1. School of Computer Science and Technology, Dalian University of Technology, Dalian, China
2. School of Software, Dalian University of Technology, Dalian, China
Email: xyliu@dlut.edu.cn

Hongfei Lin[1] and Liguo Zhang[2]
1. School of Computer Science and Technology, Dalian University of Technology, Dalian, China
2. School of Software, Dalian University of Technology, Dalian, China
Email: hflin@dlut.edu.cn

***Abstract*—Link weighting is crucial for performance of link-based web page ranking algorithms. Typically, a link is viewed as recommendation between pages, which is an unquantifiable term and existing approaches lack physical interpretations. In this paper, we view a link as the attractive force between pages, and map concepts of web (in/out degree, content similarity, etc.) to those of physics (mass, distance, attractive force). Inspired by Reilly's Law of Retail Gravitation, we propose a gravitation-like model for calculating the attractive force. We then implement a instance of our algorithm framework by taking some features of web pages into consideration. Experimental results show that this instance outperforms other typical algorithms (HITS, Randomized-HITS, and SALSA) with higher precision, better resistibility of TKC effect and no need of filtering intra-domain links.**

***Index Terms*—Link Analysis, Attractive Force Model, Link Weight, Web Information Retrieval**

I. INTRODUCTION

In modern web search engines, link-based ranking algorithms play an important role [1]. It is clear that different links are of different importance in contributing to the rank of a page on a given query. Thus a problem arises on how to assign weights to links. Till now, there is not a satisfying approach to the problem of weighting links since many features are involved, such as text in documents, link anchors, user feedback.

Typically, existing link analysis algorithms are based on the assumption that a link stands for a recommendation of one page to another. However, the term "recommendation" is hard to quantify, i.e., it is hard to determine the extent of recommendation from one page to another. This hardness essentially answers for the difficulty of weighting links and existing approaches lack physical interpretations.

The angle of view of recommendation lies at the start point of each link: a link from page A to page B exists is because that the author of A thinks B is important and makes a recommendation of B. However, from another angle of view, i.e., at the end point of each link, we can say that a link pointed to page B by page A is because that the importance of B causes the attention of A. From this point, instead of recommendation, a link can be interpreted as an attractive force between two pages, which is a quantifiable term. Note that web is a social phenomenon, and it has been observed by W.J.Reilly that the attractive force between cities follows Sir Isaac Newton's theory of gravitation (Reilly's Law of Retail Gravitation)[2], it is convincing to assume that the attractive force between pages follows a similar law.

From the above observations, in this paper, we propose a new framework for query-dependent link analysis in which each link is weighted by calculating the attractive force between its two associated pages. A mapping is built from concepts of web pages (in/out degree, content similarity, etc.) to those of physics (mass, distance, attractive force), and the attractive force is computed with formulas like theory of gravitation.

It is encouraging that we can really benefit from the nature since we have implemented an instance of the framework. Though only partial features are involved in this instance, the instance, which we call G-HITS, shows great advantages over other typical approaches such as HITS[3], Randomized-HITS[4] and SALSA[5]. Experimental results show that our instance algorithm outperform other algorithms in three aspects. 1) Better precision: G-HITS perform much better than other algorithms under the P@10 precision metric. 2) Resistance of TKC Effect: G-HITS is less vulnerable to the TKC effect than algorithms. 3) No need of filtering intra-domain links: Other algorithms need to filter intra-domain links to alleviate bias caused by mutually reinforcing effect. This is time consuming and may manslaughter some useful information. G-HITS without filtering intra-domain links perform as well as filtering, or even better.

The rest of the paper is organized as follows. Section 2 presents the background and some related works. The

Manuscript received December 28, 2010; revised March 1, 2011; accepted March 28, 2011.
Corresponding author: Xinyue Liu

doi:10.4304/jcp.7.1.124-129

attractive force model and the algorithm framework presented in section 3. In section 4 we implement a instance algorithm. The experiments and evaluations are given in section 5. Finally, we conclude the paper in section 6.

II. Background and Related Work

A. Background

Kleinberg [3] proposed that web documents had two important properties, called hubness and authority, as well as a mechanism to calculate them. In his Hyperlink-Induced Topic Search (HITS hubness) approach to broad topic information discovery, the score of a hub (authority) depended on the sum of the scores of the connected authorities (hubs). Kleinberg calculated these scores on the subset of the web that included top-ranked pages for a given query, plus those pages that pointed to or were referenced by that set.

Page and Brin [6] proposed an alternative model of page importance, called the random surfer model. In that model, a surfer on a given page i, with probability $(1 - d)$ chooses to select uniformly one of its outlinks, and with probability d to jump to a random page from the entire web. The PageRank score for node i is defined as the stationary probability of finding the random surfer at node i. PageRank is a topic-independent measure of the importance of a web page, and must be combined with one or more measures of query relevance for ranking the results of a search.

B. Related Works

Ideally, HITS is a good model for query-dependent page ranking. However, in the real web world, there are many negative characteristics that affect precision of ranking algorithms. Thus many works try to improve over HITS. Lempel and Moran [5] defined a tightly-knit community (TKC) as a small but highly connected set of sites. Even though such a community is not quite relevant to the query, it may still be ranked highly by link-based ranking algorithms. The authors proposed SALSA, a stochastic approach for link structure analysis, and demonstrated that it is less vulnerable to the TKC effect than HITS. Ng et al. [4] presented randomized HITS and subspace HITS algorithms to enhance the stability of the original HITS algorithm, etc. Though all of these algorithms adopt weighting links to address the problems, none give a clear answer or physical interpretation on why the links should be weighted in such a way. Bharat and Henzinger [7] proposed a number of improvements to HITS. One of the changes is an algorithm called imp, which re-weights links involved in mutually reinforcing relationships and drops links within the same host. They found that imp made a significant improvement over the original HITS. Chakrabarti et al. [8] extend HITS by increasing the weights of links whose anchor text incorporates terms from the query.

There are several works that try to provide general framework for links analysis. HITS emphasizes mutual reinforcement between authority and hub webpages, while PageRank emphasizes hyperlink weight normalization and web surfing based on random walk models. Ding et al [9] systematically generalized and combined these concepts into a unified framework. Xi et al [10] proposed a unified link analysis framework, called "link fusion", which considers both the inter- and intra-type link structure among multiple-type inter-related data objects and brings order to objects in each data type at the same time. Chen et al [11] proposed a unified framework that put both explicit and implicit link structures under a framework.

Finally, we note that physical models, especially gravitation model, have been successfully applied in related fields such as information retrieval and data mining. Shi et al [12] provided a gravitation-based model and bring explicit physical interpretation to formulas and concepts in information retrieval. Zhang et al [13] proposed a mechanical algorithm for data clustering which give explicit descriptions for movements of data points when falling into their genuine clusters.

III. Attractive Force Model and Algorithm Framework

A. Basic Concepts

1. Particle, Hub Mass and Authority Mass

In our model, the web pages are depicted as particles. Following Kleinberg [3], who associated every page with a hub weight and an authority weight, we associate every web page with a hub mass and an authority mass. Note that hub mass and authority mass are both query-dependent. Intuitively, a page with a high authority mass tends to attract more attention and a page with a high hub mass tends to convey more authority information. Thus, features like page content, page-query similarity and web-logs are factors of authority mass and hub mass.

We should note the more a page link to or linked by other pages, the more it can cause attention or convey information. Thus in-degree and out-degree are both main factors of hub/ authority mass.

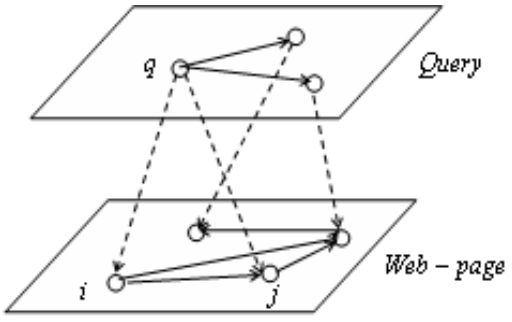


Figure 1. Query space and page space

2. Distance between Pages

The distance between two pages is a similarity metric. The WWW can be seen as a huge space, query and web page can be seen as two dimensions of the space. Most of the web link analysis research only analyzes the

hyperlinks within web pages, without considering the query dimension. However, in the real world, web pages and queries have close relationships. For example, see Fig 1, a query in the *Query* space can not only refer to other queries, but also refer to the web pages in the *Web-page* space.

3. Attractive Force between Two Pages

Assume there are two pages *i* and *j*, whose hub masses and authority masses are denoted by $M_{i,h}, M_{i,a}$ and $M_{j,h}, M_{j,a}$ respectively. There distance is denoted by $r_{i,j}$. Note that hub mass acts for conveying information and authority mass acts for attract attention, thus the attractive force between *i* and *j*, denoted by $P_{i,j}$ is dominated by $M_{i,h}$ and $M_{j,a}$, following a formula like theory of gravitation.

$$P_{ij} = \begin{cases} C\dfrac{M_{ih} \times M_{ja}}{r_{i,j}^{\ 2}}, & if \quad (i,j) \in E \\ 0, & otherwise \end{cases} \tag{1}$$

Where C is a constant. *E* is the set of the directed edges of the web graph *G= (V, E)*, derived from the given query.

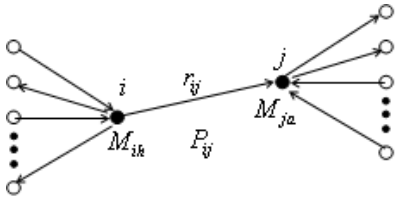


Figure 2. The attractive force between two pages

B. Algorithm Framework

1. Constructing a Sub Graph from the WWW

For a parameter *t*, the *t* highest-ranked pages for the query from a text-based search engine are collected. They are referred as the root set R_σ . The procedure is the same as HITS [3] algorithm. Then, the base set S_σ is obtained by growing R_σ to include any page pointed to by a page in R_σ and any page that points to a page in R_σ , with the restriction that every page in R_σ can bring at most *d* pages pointing to it into S_σ . So the collection of the hyperlinked pages in the S_σ can be viewed as a directed graph *G= (V, E)*.

2. Computing Hub Weights and Authority Weights

Like other typical link analysis algorithms, our algorithm framework is also based on the random walk process of Markov chains. Let there be a random surfer who follows hyperlinks in both the forward and backward directions. More precisely, the surfer starts from a randomly chosen page, and visits a new web page at every time step. Every time step, he tosses a coin with probability ε , and if the coin lands heads, he jumps to a new web page chosen at random. If the coin lands tails, then he checks if it is an odd time step or an even time step. If it is an odd time step, then he follows a randomly chosen forward link from the current page; if it is an even time step, then he traverses a random backward link of the current page. Thus, the random surfer alternately follows links in the forwards and backwards directions, and occasionally "resets" and jumps to a page chosen at random. That can be expressed in the matrix-vector terms as follows:

$$a^{(t+1)} = \varepsilon\vec{1} + (1-\varepsilon)P_r^T \times h^{(t)} \tag{2}$$

$$h^{(t+1)} = \varepsilon\vec{1} + (1-\varepsilon)P_c \times a^{(t+1)} \tag{3}$$

Where *P* is the adjacency matrix derived from the query-specific link graph, each element of the matrix *P* is the value of the attractive force that one page links to the other. $\vec{1}$ is the vector consisting of all ones, P_r is the same as *P* with its rows normalized to sum to 1, and P_c is *P* with its columns normalized to sum to 1.

IV. An Algorithm Instance

Implementation of our algorithm framework need concrete methods for computing masses and distances. In this section, we provided an instance of the algorithm in which part of the features are involved.

A. Computing hub mass and authority mass

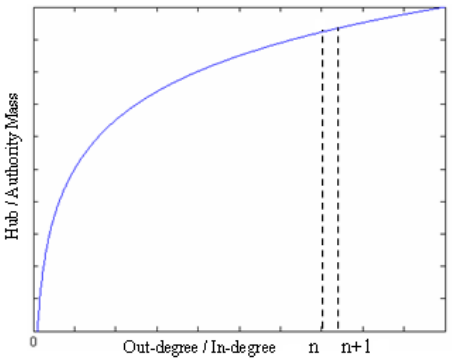


Figure 3. Increasing trends of hub/authority mass, with the increasing of the out-degree / in-degree

It has been analyzed that in-degree and out-degree are both main factors of hub/ authority mass. Let investigate how they affect the masses. Intuitively, the more a page is pointed by other pages, the higher authority, and it should be assigned a lager mass. In the beginning, when a page *i* has a small number of in-links, adding a new link to it is of much importance for it to gain more attention. However, when a page *i* already has a large number in-links, a new added link has little impact to its popularity. This means that a page's authority mass increase with its in-degree following a curve like Fig 3. The relationship between hub mass and out-degree is similar. We adopt logarithm functions to describe them. Note that out-degree also affect authority mass, since a page link to many other tends to be known more, but its impact is much less than in-degree. The relationship between authority mass and in-degree is similar.

The formulas of hub mass M_{ih} and authority mass M_{ia} are as follows. They are the base-e logarithm functions of $|F(i)|$ and $|B(i)|$, which represent out-degree and in-degree respectively.

$$M_{ih} = \alpha \times \ln(|F(i)|+1) + \beta \times \ln(|B(i)|+1) \quad (4)$$

$$M_{ia} = \alpha \times \ln(|B(i)|+1) + \beta \times \ln(|F(i)|+1) \quad (5)$$

Where α and β are the dependent factors. We set the factors as follows: α is 0.80-0.85, β is 0.15-0.20 and $\alpha + \beta = 1$.

B. Computing distance between pages

According to the relationship of queries and web pages in the space, distance $r_{i,j}$ between two pages can be reflected by the distance of pages *i* and *j* to query *q*, which can be indirectly reflected by the content similarities of pages *i* and *j* to query topic *q*. That can be expressed as follows.

$$r_{i,j} = \lambda \times f(s_{q,i}, s_{q,j}) \quad (6)$$

Where λ is the dependent factor and satisfies $0 < \lambda \le 1$. If λ is larger, the distance will be dependent on the content similarities more. The formula of content similarity $s_{q,j}$ of page *j* to the query topic *q* is showed in the below, $s_{q,i}$ is the same with it.

$$s_{q,j} = \frac{\sum_{k=1}^{t} w_{k,j} \times w_{k,q}}{\sqrt{\sum_{k=1}^{t} w_{k,j}^2} \times \sqrt{\sum_{k=1}^{t} w_{k,q}^2}} \quad (7)$$

Where $w_{k,j} = tf_{k,j} \times idf_k$, $w_{k,q} = tf_{k,q} \times idf_k$, and $tf_{k,j}$ is the frequency of the term *k* in document *j*, $tf_{k,q}$ is the frequency of the term *k* in query *q*, idf_k is an estimate of the inverse document frequency of the term *k* on the WWW.

Then, the formula of the distance is as follows.

$$r_{i,j} = \lambda \times (1/\sqrt{s_{q,i}^{\ 2} + s_{q,j}^{\ 2}}) \quad (8)$$

Formula (8) also reflects distance of page *i* to page *j* reasonably, we can assume that query *q*, page *i* and *j* can form a right-angled triangle in the data space. $r_{i,j}$, $s_{q,i}$ and $s_{q,j}$ are the edges of the triangle. The physical meaning of distance $r_{i,j}$ between two pages is that if content similarities $s_{q,i}$ and $s_{q,j}$ are larger, the distance of page *i* to *j* will be nearer.

C. The algorithm

The algorithm is a HITS-like algorithm with a gravitation model. We name it G-HITS. The algorithm is described as follows.

Algorithm 1: G-HITS algorithm to compute the hub weights and authority weights

Input: The sub graph *G (V, E)* derived from the given query
Output: *h*, *a*
Iteration (*G*)
F (i) denote the set of all pages *i* points to *i*
B (i) denote the set of all pages pointing to *i*
P_{ij} denote the attractive force from *i* to *j*

1. Let $z = [1, 1, \ldots, 1]^T$
2. Initialize *h* and *a*, set $h = z$, $a = z$
3. $h_i = \varepsilon \vec{1} + (1-\varepsilon) \sum_{j \in F(i)} a_j \times \frac{1}{|B(j)|} \times P_{ij}$
4. $a_i = \varepsilon \vec{1} + (1-\varepsilon) \sum_{j \in B(i)} h_j \times \frac{1}{|F(j)|} \times P_{ji}$
5. Normalize *h* and *a*
6. Obtaining $h_i = h_i / \|h\|$, $a_i = a_i / \|a\|$
7. Repeat 3, 4, 5, and 6 until convergence

Return *h*, *a*

The time complexity of the iteration algorithm above is $O(N^2)$. The element P_{ij} can be expressed as follows:

$$P_{ij} = f(M_{ih}, M_{ja}, r_{i,j}) \times W_{ij} \quad (9)$$

Where
$$W_{ij} = \begin{cases} 1, & if \ (i,j) \in E \\ 0, & otherwise \end{cases} \quad (10)$$

According to the Randomized HITS algorithm [10], the iteration will cause $a^{(t)}$ and $h^{(t)}$ converge to the odd-step and even-step stationary distributions.

V. Experiments and Evaluations

A. Experimental Methods

TABLE I.
THE SIMILARITY THRESHOLDS OF THE QUERIES

Queries	abortion	jaguar	geometry	genetic	bicycling	java
$s_{q,iMin}$	1.0E-06	5.0E-07	5.0E-07	5.0E-07	5.0E-07	1.0E-06

TABLE II.
THE DATASET DETAIL FOR EACH QUERY, FILTERING THE INTRA-DOMAIN URLS

Queries	abortion	jaguar	geometry	genetic	bicycling	java
B_σ size	1546	1097	2185	2117	1892	2851
Links	1018	1011	2190	2024	4327	9290

We performed experiments on G-HITS and three other typical algorithms with the following 6 different queries: abortion, jaguar, geometry, genetic, bicycling and java, which have been used in previous works [3]. We first obtained the root set R_σ and base set s_σ for every query like HITS [3] algorithm, and then we got the directed graph *G= (V, E)* derived from the s_σ .Secondly, we attempted to filter the noninformative URLs, which exist in the web pages, according to the distributions of the content similarities of the pages for each query. We set a similarity threshold $s_{q,iMin}$ for each query, showed in Table

1. Then we filtered the URLs whose content similarities are below $s_{q,iMin}$, and we got the new base set B_σ. Thirdly, we did the experiments under two conditions as follows:

(1) Filtering intra-domain links: In this experiment, we filtered intra-domain links, following HITS and SALSA algorithms. The size and the links of B_σ are presented in Table 2.

(2) No filtering the intra-domain links: In this experiment, we didn't filter the intra-domain links, since we want to have a comparison with the filtered experiments. The size and the links of B_σ are presented in Table 3.

Then, we set the factors α =0.80, β = 0.20, λ =0.95, C=1.00, t=200, d=50, the convergence threshold δ to be 1.0E-06, and the jump probability ε to be 0.20 for every query.

TABLE III.
THE DATASET DETAIL FOR EACH QUERY, WITHOUT FILTERING THE INTRA-DOMAIN URLS

Queries	abortion	jaguar	geometry	genetic	bicycling	java
B_σ size	1546	1097	2185	2117	1892	2851
Links	9536	9494	6704	11385	22846	49538

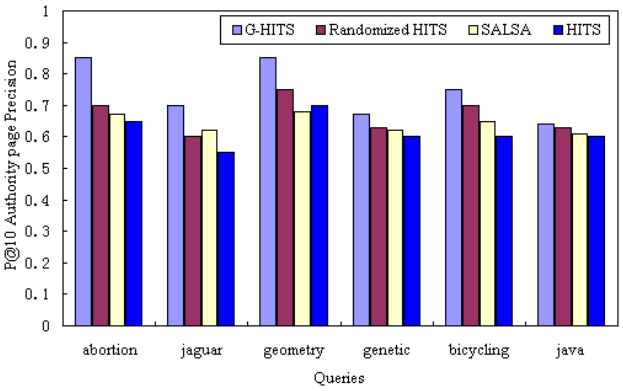


Figure 4. Comparison of P@10 Authority Page Precision of the G-HITS, Randomized HITS, SALSA, and HITS, filtering the intra-domain links

B. Results Analysis and Evaluations

1. Authorities Evaluation

P@10（Precision at 10）is the number of relevant documents in the top 10 documents in the ranked list returned for a topic. If the document is relevant, the score of it will be 1; else it will be 0.

We asked 100 volunteers to evaluate the result authorities according to our rules above. Then we got the average high relevant rate for each query topic, marked as P@10 Authority Page Precision. We can see the performance comparison of the four algorithms in Fig 4, the experiment has filtered the intra-domain URLs. And in Fig 5, the experiment is without filtering intra-domain links.

(1) The experiment with filtering intra-domain links.

We performed this experiment following HITS, SALSA and other algorithms to filter the intra-domain URLs. We can see that G-HITS algorithm get higher authorities than other three typical link analysis algorithms, in appendix 1 and 2. For example, the results authorities of Top1-Top10 for the query "abortion", showed in appendix 1. Seven URLs are quite popular with the volunteers in G-HITS algorithm, which are Top1 to Top6, and Top10. But there are four, five, and five URLs are popular, in HITS, SALSA, and Randomized HITS algorithms, respectively.

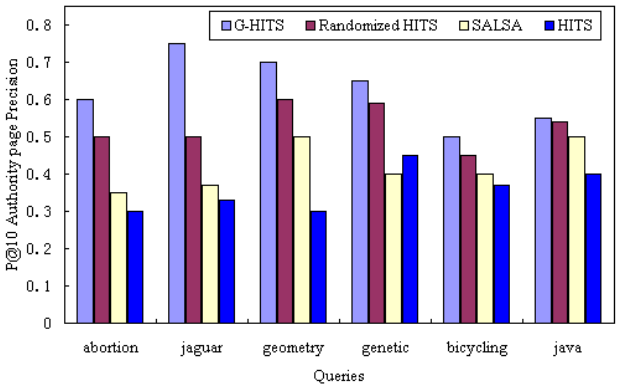


Figure 5. Comparison of P@10 Authority Page Precision of the G-HITS, Randomized HITS, SALSA, and HITS, without filtering the intra-domain links

(2) The experiment without filtering intra-domain links.

We can see the performance of different result authorities in appendix 3, with the query "jaguar". The authorities of G-HITS are much better than other three algorithms. We can compare the experimental results of the filtered and unfiltered intra-domain links in appendix 2 and appendix 3, with the same query "jaguar". We find that the unfiltered results of G-HITS in appendix 2 are even better than the filtered in appendix 3. But for the other three algorithms, it can be seen that the filtered results are better than unfiltered. Thus, we can get rid of the redundant time of filtering. This is because that G-HITS has strong resistibility against TKC effect, which is explained in the following sub-sections.

2. Resistance of TKC Effect

The TKC Effect occurs in HITS algorithm for the query topic "jaguar", showed in appendix 2, which has filtered intra-domain links. It is obviously that Top6 to Top10 are URLs from the same domain. That is the result of the Mutual Reinforcement Approach of HITS algorithm. Top8 to Top10 are same domain URLs of SALSA algorithm, which is less vulnerable than HITS in the TKC Effect. Randomized HITS also has TKC effect obviously. But see the authorities of G-HITS algorithm presented in appendix 2, we find that just Top4 and Top5 are in the same domain. Then, let's see appendix 3, which is results without filtering intra-domain links. The results of HITS, SALSA and Randomized HITS are quite bad because of the TKC Effect occurred. But the results of G-HITS are much better than them, and show the robustness against the TKC Effect. Similar phenomena are also found in the top ten authorities of query "genetic", "bicycling" etc. Thus, G-HITS algorithm is more resistant

to the TKC effect than other algorithms, no matter on the condition of the filtering intra-domain links, or no filtering intra-domain links.

We can give brief analysis to this phenomenon. In G-HITS algorithm, different links correspond to different attractive force values from physical perspective. For example, web page *k* links to both *i* and *j*, if the content similarity of page *i* to the given query *q* is larger than *j*, the distance of page *k* to *i* will be nearer than *k* to *j*; at the same time, if the authority mass of page *i* are not the same with the page *j*, then, attractive force P_{ki} will be different from P_{kj} .Thus, the mutually reinforcing effect alleviated and no page can benefit too much from its neighbor, which can lead to TKC Effect. While at the same time, not all intra-domain links are used for navigation, some are also used for recommendation. This explains why no filtering of intra-domain links for G-HITS sometimes performs better than filtering.

VI. Conclusions and Future Work

In this paper, an attractive force model for link weighting is proposed from the perspective of social science inspired by Reilly's Law of Retail Gravitation. An algorithm framework for base query-dependant web page ranking is then deduced and an instance of the framework is implemented. Experimental results show that it is encouraging that we can really benefit from the nature. An open problem is that how can we fuse all the features that affect page ranking into this model. Another problem is how we can develop a similar model for query-independent web page ranking.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 60873180 and the Fundamental Research Funds for the Central Universities.

References

[1] Allan Borodin et al, "Link Analysis Ranking Algorithms Theory and Experiments," *ACM Transactions on Internet Technologies*, 2005, vol. 5, pp. 231-297.

[2] William John Reilly, "The law of retail gravitation," *New York*, W. J. Reilly, 1931.

[3] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, 1999, vol. 46, pp. 604-632.

[4] A.Y.Ng, A.X.Zheng, and M. I. Jordan, "Stable Algorithms for Link Analysis," *Proc. 24th International Conference on Research and Development in Information Retrieval*, *New Orleans*, *Louisiana*, *USA*, 2001.

[5] R. Lempel and S. Moran, "The Stochastic Approach for Link-structure Analysis (SALSA) and the TKC effect," *In: Proceeding 9th International World Wide Web Conference*, *Amsterdam, Netherlands*, 2000.

[6] S. Brin and L. Page, "The Anatomy of a Large-scale Hyper textual Web Search Engine," *Computer Networks and ISDN Systems*, 1998, vol. 30, pp. 107-117.

[7] K. Bharat and M. R. Henzinger, "Improved algorithms for topic distillation in hyperlinked environments," *In Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, pp. 104–111.

[8] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, S. Rajagopalan, "Automatic resource compilation by analyzing hyperlink structure and associated text," *Proceedings of the 7th ACM-WWW International Conference, Brisbane*, 1998, pp. 65-74.

[9] Chris Ding, Xiaofeng He, Parry Husbands, Hongyuan Zha and Host D. Simon, "PageRank, HITS and a unified framework for link analysis," *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 353-354.

[10] Wensi Xi, Benyu Zhang, Yizhou Lu, Zheng Chen, Shuicheng Yan, Huajun Zeng, Wei-Ying Ma, and Edward A. Fox, "Link Fusion: A Unified Link Analysis Framework for Multi-Type Interrelated Data Objects," *The Thirteenth World Wide Web conference (WWW 2004)*, 2004, pp. 203-211.

[11] Zheng Chen, Li Tao, Jidong Wang, Liu Wenyin, Wei-Ying Ma, "A Unified Framework for Web Link Analysis," *Proceedings of the ACM Workshop Wireless Security*, 2002, pp. 63-72.

[12] Shuming Shi, Ji-Rong Wen, Qing Yu, Ruihua Song, Wei-Ying Ma, "Gravitation-Based Model for Information Retrieval," *The 28th Annual International ACM SIGIR Conference (SIGIR'2005)*, 2005.

[13] Xianchao Zhang, He Jiang, Xinyue Liu and Hong Yu, "A Clustering Algorithm Based on Mechanics," *Lecture Notes in Artificial Intelligence*, 2007, vol. 4426, pp. 367-378.

Xinyue Liu received the M.S degree in Computer Science and technology from Northeast Normal University, China, in 2006. She is currently working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. Her research interests include multimedia information retrieval, web mining and machine learning.

Hongfei Lin received the Ph.D degree from Northeastern University, China. He is a professor in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. His professional interests lie in the broad area of information retrieval, web mining and machine learning, affective computing.

Liguo Zhang received his M.S. degree in Software Engineering from Dalian University of Technology in 2008. His major interests lie in web link analysis.

An Improved HITS Algorithm Based on Page-query Similarity and Page Popularity

Xinyue Liu[1,2]
1. School of Computer Science and Technology, Dalian University of Technology, Dalian, China
Email: xyliu@dlut.edu.cn

Hongfei Lin[1] and Cong Zhang[2]
2. School of Software, Dalian University of Technology, Dalian, China
Email: hflin@dlut.edu.cn

***Abstract*—The HITS algorithm is a very popular and effective algorithm to rank web documents based on the link information among a set of web pages. However, it assigns every link with the same weight. This assumption results in topic drift. In this paper, we firstly define the generalized similarity between a query and a page, and the popularity of a web page. Then we propose a weighted HITS algorithm which differentiates the importance of links with the query-page similarities and the popularity of web pages. Experimental results indicate that the improved HITS algorithm can find more relevant pages than HITS and improve the relevance by 30%-50%. Furthermore, it can avoid the problem of topic drift and enhance the quality of web search effectively.**

***Index Terms*—HITS Algorithm, Link Analysis, Similarity, Popularity**

I. INTRODUCTION

With the rapid growth of computer technique and the Web, 2EB (1EB≈1024*1024*1024GB) information is produced each year in the whole world and the increasing speed of information has been beyond imagination. Thus, how to effectively search relevant information from the huge information on the internet is the primary goal of the modern search engines. Link analysis plays an important role in accomplishing this task.

The most two famous link analysis algorithms are PageRank algorithm [1] by Sergey Brin and Lawrence Page and the HITS algorithm [2] by Kleinberg. They are all purely link-based algorithm which do not think about the content of the page so that resulting in the problem of topic drift. That means the results of the algorithms often contain some non-relevant pages with tightly interconnected density. In order to control the topic drift, page ranking algorithms based on hyperlinks and content have been proposed, such as the ARC algorithm [3] and the Average algorithm [4]. HITS algorithm can not treat links differently, due to its definition makes the quality of hubs is determined by it pointed to the quantity of authority page. In 2001, Allan Borodin and others proposed the Hub-Averaging (HUBAVG) algorithm which sets the hub weight of some node *i* to the average authority weight of the authorities pointed to by hub *i*. Later, Allan Borodin and others proposed the Authority-Threshold algorithm [5] which sets the hub weight of node *i* to be the sum of the *k* largest authority weights of the authorities pointed to by node *i*. This corresponds to saying that a node is a good hub if it points to at least *k* good authorities. Moreover, Lempel and Moran proposed the SALSA algorithm [6] based on the Markov chain. Cohn and Chang proposed the PHITS algorithm [7] based on the probability model, and so on.

Based on the HITS algorithm, this paper finds that the HITS algorithm treats links equally and does not take use of the content of the page when distributes rank scores. As a result, the definition of the similarity of the pages and the popularity of the pages are proposed by this paper to influence the rank result. This paper proposes the Improved HITS (I-HITS) algorithm based on similarity and popularity, which differentiates the importance of links with the similarity of pages and the query topic and the popularity of pages. And I-HITS algorithm constructs a new adjacency matrix to compute hubs and authorities. Theoretical analysis and experimental results both show that the I-HITS algorithm performances better in search precision and avoids the problems of topic drift effectively.

The rest of this paper is organized as follows. A brief background review of HITS algorithm and the problems it has been presented in Section 2. An extended HITS algorithm, the Improved HITS algorithm is described in Section 3. Experimental results of I-HITS are given in Section 4. Section 5 summarizes the conclusions.

II. HITS ALGORITHM AND PROBLEM ANALYSIS

A. HITS algorithm

Unlike PageRank algorithm, Kleinberg proposed a more refined notion for the importance of the Web pages. He believed that the importance of pages is dependent with the query topic. In his framework of Fig. 1, every page can be considered as two identities, hub and authority. The link structure can be described as a

Manuscript received December 28, 2010; revised March 1, 2011; accepted March 28, 2011.
Corresponding author: Xinyue Liu

doi:10.4304/jcp.7.1.130-134

dependent relationship: A good authority is a page pointed to by good hubs, while a good hub is a page that points to good authorities. Therefore, Kleinberg defined the following mutual reinforcing relationship between hubs and authorities: the hub weight to be the sum of the authorities of the nodes that are pointed to by the hub, and the authority weight to be the sum of the hub weights that point to this authority [2] [8] [9].

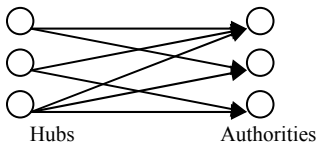


Figure 1. Hubs and authorities

$$a_i = \sum_{j \in B(i)} h_j \quad \text{and} \quad h_i = \sum_{j \in F(i)} a_j \tag{1}$$

Let a_i and h_i represent the authority and hub weight of page i, respectively. $B(i)$ and $F(i)$ denote the set of referrer and reference pages of page i, respectively.

B. The problems of the HITS algorithm

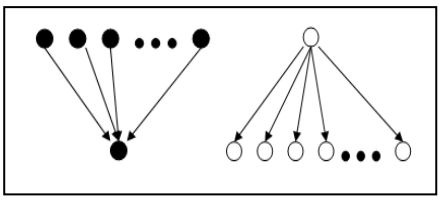


Figure 2. A bad example for HITS

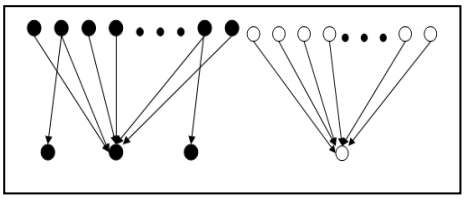


Figure 3. Another bad example for HITS

The definition of HITS algorithm has the following two implicit properties. It is symmetric, in the sense that both hub and authority weights are defined in the same way. If we reverse the orientation of the edges in the graph, authority and hub weights are swapped. The HITS algorithm is also egalitarian, in the sense that when computing the authority weight of some page p, the hub weights of the pages that point to page p are all treated equally (same with computing the hubs weights). However, these two properties may sometimes lead to non-intuitive results. Consider the example graph in Fig. 2 demonstrates that if the number of white authorities is larger than the number of black hubs, the HITS algorithm will allocate all authority weight to the white authorities, while giving little weight to the black authority and easily cause topic drift. However, intuition suggests that the black authority is better than the white authorities and should be ranked higher. Similarly, in Fig. 3, after computing, the middle black authority will have higher authority weight than the white authority, but actually they should be equally good. Therefore, we seek to change the symmetric and the egalitarian of the HITS algorithm, and aim at treating links differently.

III. I-HITS ALGORITHM

In view of the problem of the HITS algorithm, the paper proposes the I-HITS algorithm, which is according to the product of the relevance of Web page and query topic, the relevance of Web page linked to and query topic and the popularity of the page to weight the link instead of the average transfer strategy.

A. Definition 1: The similarity of the page and the query

If page i points to page j, then i is called the source page and j is called the target page. In most cases, the more information a source page contains with the query Q, this source page is more relevant with the query topic, and the similarity is S_i. Similarly, the more information a target page contained with the query Q, this target page is more relevant with the query topic as well, and the similarity is S_j. Furthermore, we find that anchor text is used to describe the target document, not to describe the current document [1] [10], and it summarizes the topic of the target document with a high degree of accuracy [11]. Therefore, in order to reduce the computational complexity, computing the similarity of the target page and the query Q is simplified by computing the similarity of the anchor text and the query Q. Thus, in the adjacency matrix, if page i points page j, the item in it is $(1+ S_i)*(1+ S_j)$; if not, it is 0.

B. Definition 2: The popularity of the page

The more popular a page is, the more other pages tend to point to it or it will be linked to by other pages. The proposed extended I-HITS algorithm contributes larger rank values to more important (popular) pages instead of dividing the rank value of a page evenly among its outlink pages.

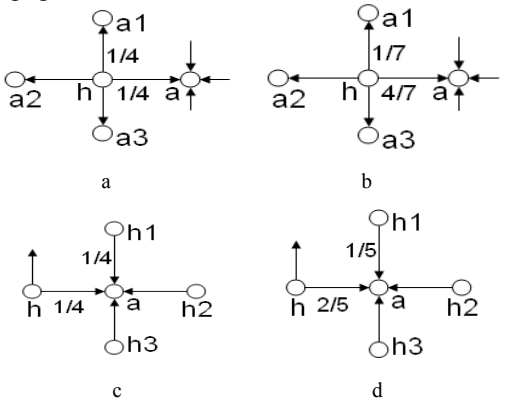


Figure 4. How to distribute weight with popularity

Fig. 4.a represents the HITS algorithm. According to the out-degree of h, h allocates its weight as the probability of a quarter. That is, a, a_1, a_2 and a_3 will get the same value of hub. However, in Fig. 4.b, with the definition of the popularity, the hub value of each webpage is determined by the proportion of its in-degree

pointed to the overall in-degree. That is, allocating hub weights according to the migration probability of links. The in-degree of *a*, a_1, a_2, a_3 are 4, 1, 1, 1, then *a* gets 4/7 of the hub weight and a_1, a_2, a_3 gets 1/7. Similarly, in Fig. 4.c, when HITS computing authority values, the chances of the authority pages accumulating the authority values are equal. All the *h*, h_1, h_2, h_3 pointing to "*a*" will have the same chance of gaining authority values. In the I-HITS algorithm, the authority value of each webpage pointing to "*a*" is determined by the proportion of its out-degree to the overall out-degree, as shown in Fig. 4.d. E.g. with the respective out-degree values of *h*, h_1, h_2, h_3 at 2、1、1、1, the authority values of *h*, h_1, h_2, h_3 are 2/5, 1/5, 1/5, 1/5 respectively. Advantage of this way is that allocating more rank values to the most popular pages from an objective point of view. Like in a certain area, the more fans he has, more famous he should be.

The popularity from the number of outlinks is recorded as $W_{(j,i)}^{out}$ [12]. $W_{(j,i)}^{out}$ is calculated based on the number of outlinks of page *i* and the number of outlinks of all reference pages of page *j*. Where O_i and O_p represent the number of outlinks of page *i* and page *j*, respectively. R(*j*) denotes the reference page list of page *j*.

$$W_{(j,i)}^{out} = \frac{O_i}{\sum_{p \in R(j)} O_p} \tag{2}$$

The popularity from the number of inlinks is recorded as $W_{(j,i)}^{in}$ [12]. $W_{(j,i)}^{in}$ is calculated based on the number of inlinks of page *i* and the number of inlinks of all reference pages of page *j*. Where I_i and I_p represent the number of inlinks of page *i* and page *j*, respectively. R(*j*) denotes the reference page list of page *j*.

$$W_{(j,i)}^{in} = \frac{I_i}{\sum_{p \in R(j)} I_p} \tag{3}$$

In this example of Fig. 5, page A points page C and page D. The inlinks and outlinks of these two pages are I_C=2, I_D=1, O_C=2, O_D=3. Therefore, $W_{(A,C)}^{out} = O_C / (O_C + O_D)$ =2/5 and $W_{(A,C)}^{in} = I_C/(I_C + I_C)$ =2/3.

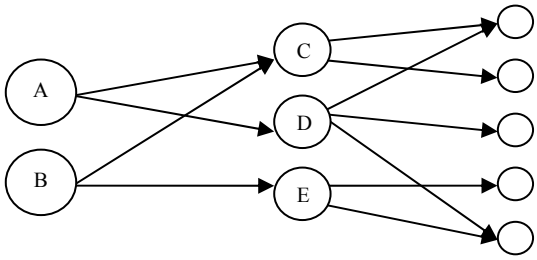


Fig 5. Links of Web

C. The I-HITS algorithm

(1) Assign each page with two value: a_i and h_i, represent authority weight and hub weight.

(2) Initialize: a_i=1, h_i=1(i=1, 2, 3, … , n).

(3) Compute:

$$a_i = \sum_{j \in B(i)} h_j * (1 + s_i) * (1 + s_{ji}) * \frac{I(i)}{\sum_{p \in F(j)} I(p)}$$

$$h_i = \sum_{j \in F(i)} a_j * (1 + s_i) * (1 + s_{ij}) * \frac{O(i)}{\sum_{p \in B(j)} O(p)} \tag{4}$$

Normalization.

(5) If a_i and h_i do not converge, turn to step (3).

IV. EXPERIMENTS

A. Evaluation method

TREC (text retrieval conference) is the greatest impact of evaluation conference in the text information retrieval areas. This experiment uses TREC's P@10 evaluation criteria. That is, P@10 is the number of relevant documents in the top 10 documents in the ranked list returned for a topic.

TREC usually use binary and ternary evaluation. This experiment uses a more specific and detailed ternary evaluation and classifies a document as:

(1) Highly relevant (HR): Contain very important and very authoritative information about the given query.

(2) Relevant(R): Have relevant but not important information about the given query.

(3) Non-relevant (NR): Include neither the keywords of the given query nor relevant information about it.

An objective categorization of the results is achieved by integrating the responses from several people who are able to offer impartial opinions: for each page, we compared the count of each category and chose the category with the largest count as the type of that page.

B. Experiment data

In order to evaluate the I-HITS algorithm, this experiment compared with HITS and I-HITS on 5 different queries, four in English and one in Chinese: abortion, alcohol, basketball, movies, 搜索引擎. All of these queries have already appeared in previous works [2] [13] [14] and are representative. The base sets for these queries are constructed in the fashion described by Kleinberg. Statistical data is in Table 1.

TABLE I.
EXPERIMENT DATA

query	nodes	hubs	authorities	links
abortion	1652	949	933	2849
alcohol	1964	1441	1213	11083
basketball	1153	930	641	3588
movies	2934	2051	1885	18210
搜索引擎	2884	2142	1744	37941

C. Evaluation

The following table shows three queries results of alcohol, movies and 搜索引擎. In these tables, the results that are labeled highly relevant appear in boldface, while the relevant ones appear in italics, and the rest are non-relevant.

Table 2 shows the results of alcohol. The top 10 results returned by HITS are all relevant, 7 of them are highly

relevant. The top 10 results returned by I-HITS are all relevant, 8 of them are highly relevant, the proportion of highly relevant improves by 10%.

TABLE II.
RESULTS OF ALCOHOL

rank	HITS	I-HITS
1	**http://www.niaaa.nih.gov/**	*http://www.ndp.govt.nz/publications/review-*
2	**http://www.health.org/**	**http://www.wrap.org/**
3	**http://faculty.washington.edu/chudler/alco.html**	**http://www.alcoholmedicalscholars.org/**
4	*http://www.alcoholfreechildren.org*	**http://www.niaaa.nih.gov**
5	**http://www.nasadad.org/**	**http://www.health.org/**
6	*http://www.nofas.org/*	**http://www.alcoholconcern.org.uk/**
7	**http://ncadi.samhsa.gov/govpubs/ph323**	**http://ncadi.samhsa.gov/govpubs/ph323/**
8	**http://www.alcoholconcern.org.uk/**	**http://faculty.washington.edu/chudler/alc**
9	*http://www.atf.treas.gov/*	**http://www.cdc.gov/alcohol/**
10	**http://www.hsph.harvard.edu/nutritionso**	*http://www.atf.treas.gov/*

TABLE III.
RESULTS OF MOVIES

rank	HITS	I-HITS
1	**http://movies.yahoo.com/**	http://www.gazettenet.com/dining
2	**http://www.nytimes.com/pages/movies/in**	**http://www.onwisconsin.com/movies/**
3	**http://movies.msn.com/**	*http://us.lrd.yahoo.com/_ylt=A9FJq6Dw5x*
4	http://www.wmm.com/	**http://www.hotfreelayouts.com/layouts.p**
5	http://dmoz.org/Arts/Movies/	*http://www.bio.unc.edu/faculty/goldstein/la*
6	*http://www.moviesunlimited.com/*	**http://www.frazy.com/**
7	http://service.real.com/filmcom/	*http://www.bio.davidson.edu/Courses/movie*
8	**http://movies.about.com/**	*http://www.gnovies.com/*
9	*http://www.empiremovies.com/links.php*	**http://www.yourmovies.com.au/**
10	http://www.teachwithmovies.org/	**http://www.reelclassics.com/**

TABLE IV.
RESULTS OF 搜索引擎

rank	HITS	I-HITS
1	**http://www.google.com/intl/zh-CN/**	*http://www.xpue.net*
2	*http://www.gseeker.com*	*http://www.tooooold.com*
3	http://www.wangtam.com/50226711/c_wav	*http://www.bbssearch.cn*
4	http://www.yuleguan.com	**http://www.google.com/intl/zh-CN/**
5	http://www.chinaventurenews.com	http://www.1hd.cn
6	http://www.tjacobi.com	**http://www.baidu.com/**
7	http://www.money-courier.com	*http://bizsite.sina.com.cn/*
8	http://www.geekervision.com	http://it.sohu.com/7/0903/35/column213613
9	http://www.in-women.com	**http://so.163.com/**
10	http://www.tracingadget.com	http://news.qq.com/a/20070405/001271.ht

Table 3 shows the results of movies. 6 of the top 10 results returned by HITS are relevant, 4 of them are highly relevant. 9 of the top 10 returned by I-HITS are relevant, 5 of them are highly relevant. The proportion of highly relevant improves by 10% and the proportion of relevant improves by 30%.

Table 4 shows the results of 搜索引擎. 2 of the top 10 results returned by HITS are relevant, 1 of them are highly relevant. 7 of the top 10 returned by I-HITS are relevant, 3 of them are highly relevant. The proportion of highly relevant improves by 20% and the proportion of relevant improves by 50%.

From Table 2 to Table 4, to some extent, HITS algorithm and I-HITS algorithm all will product the problem of topic drift. However, Table 5 and Table 6 show that I-HITS algorithm generates more relevant pages than HITS, which illustrates I-HITS is better than HITS in the precision. Furthermore, the proportion of highly relevant improves by 10%-20% and the proportion of relevant improves by 30%-50%.

V. CONCLUSION

TABLE V.
PROPORTION OF HR

query	HITS	I-HITS
alcohol	70%	**80%**
movies	40%	**50%**
搜索引擎	10%	**30%**

TABLE VI.
PROPORTION OF R

query	HITS	I-HITS
alcohol	**100%**	**100%**
movies	60%	**90%**
搜索引擎	20%	**70%**

The research of page ranking algorithm is significant because it will increase the accuracy of search engines**.** At the beginning this article briefly introduced mainstream page ranking algorithms, especially analyzed HITS algorithm and its existing problem. Based on HITS algorithm, this article proposed I-HITS algorithm, which

is an algorithm according to parameters of similarity and popularity. I-HITS algorithm measured the importance of page link based on two page attributes, which includes source page, target page, relevancy of query topic and the popularity of page in Network subgraph. As a result, I-HITS increased the ability of distinguish link importance, and avoid top drift. Furthermore, this article compared HITS algorithm with I-HITS algorithm by experiment. The result shows that I-HITS algorithm is better than HITS algorithm when searching related page. The quality of query is increased obviously.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 60873180 and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, 1998, vol. 30, pp. 107-117.

[2] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *In Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 668-677.

[3] Chakrabarti S, Dom B, Raghavan P, etal, "Automatic resource compilation by analyzing hyperlink structure and associated text," *Computer Networks and ISDN Systems*, 1998, vol. 30, pp. 65-74.

[4] Gevrey J, Ruger S, "Link-based approaches for text retrieval," *Proceedings of TREC-10, NIST (Gaithersburg, MD, 13-16, Nov 2001), NIST Special Publication*, 2002, pp. 279-285.

[5] Borodin A, Roberts G O, Rosenthal J S, "Finding Authorities and Hubs from Link Structures on the World Wide Web," *In Proceedings of the 10th International Conference on World Wide Web, Hong Kong*, *China,* 2001, pp. 415-429.

[6] Lempel R, Moran S, "The stochastic approach for link-structure analysis (SALSA) and the TKC effect," *Computer Networks*, 2000, vol. 33, pp. 387-401.

[7] Cohn D, Chang H, "Learning to probabilistically identify authoritative documents," *In Proceedings of the 17th International Conference on Machine Learning (ICML-2000)*, *Stanford University*, *United States,* 2000, pp. 167-174.

[8] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg, "Mining the Web's link structure," *Computer*, 1999, vol. 32, pp. 60-67.

[9] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon, "Link analysis: Hubs and authorities on the world," *LBNL Tech Report 47847*, 2001.

[10] N Craswell, D Hawking, S E Robertson, "Effective site finding using link anchor information," *In Research and Development in Information Retrieval*, 2001, pp. 250-257.

[11] Zhang Min, Gao Jianfeng, Ma Shaoping, "Anchor Text and Its Context Based Web Information Retrieval," *Journal of Computer Research and Development*, 2004, vol. 41, pp. 221~226.

[12] Wenpu Xing and Ali Ghorbani, "Weighted PageRank Algorithm," *In Proceedings of the Second Annual Conference on Communication Networks and Services Research*, 2004.

[13] C. Dwork, R.Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the Web," *In Proceedings of the 10th International World Wide Web Conference*, *Hong Kong*, 2001, pp. 613-622.

[14] R. Lempel and S. Moran, "The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect," *In Proceedings of the 9th International World Wide Web Conference*, 2000, pp. 387-401.

Xinyue Liu received the M.S degree in Computer Science and technology from Northeast Normal University, China, in 2006. She is currently working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. Her research interests include multimedia information retrieval, web mining and machine learning.

Hongfei Lin received the Ph.D degree from Northeastern University, China. He is a professor in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. His professional interests lie in the broad area of information retrieval, web mining and machine learning, affective computing.

Cong Zhang received her M.S. degree in Software Engineering from Dalian University of Technology in 2008. Her major interests lie in web link analysis.

A Cooperative and Heuristic Community Detecting Algorithm

Ruixin Ma
School of software, Dalian University of Technology, Dalian, China
Email: teacher_mrx@126.com

Guishi Deng and Xiao Wang
School of Management Science and Engineering, Dalian University of Technology, Dalian, China
Email: Denggs@dlut.edu.cn, kara0807@126.com

***Abstract*—This paper introduces the concept of community seed, vector and relation matrix. In terms of the relation similarity between free vertices and the existing communities, we put vertices into different groups. A minimum similarity threshold is proposed to filter which gives a method to find the vertices located at the overlapped area between different communities. This paper analyzes a series of network dataset and proves that our algorithm is able to accurately find communities with high cohesion and weak coupling. We use a variety of test to demonstrate that our algorithm is highly effective at detecting community structures in both computer-generated and real-world networks.**

***Index Terms*—community seed; relation matrix; minimum similarity threshold; overlapped area; high cohesion and weak coupling**

I. INTRODUCTION

Network has attracted considerable recent attention as one of the most powerful mathematical presentation for complex systems. Network ideas have been applied with success to topics as diverse as scientific citation and collaboration [1], epidemiology [2], ecosystems, to name but a few. Previous study indicates that large complex networks have statistical features such as small-world phenomenon [3], isomerism [4], clustering, scale-free [5]. According to the recent research, complex networks also have obvious community structures [6]. Community structure is the reflection of networks' modularization and heterogeneity; it tells us that the real world is combined by all sorts of vertices. It is of vital significance to deeper study the community structures which hidden behind the complex networks, as well as to intellectually mine them. For example, communities in social networks can be used to find vertices of similar interests and social context; classifying the pages on world wide web is good to improve the search efficiency and accuracy, and help to implement the function of information filtering, hot spots tracking, news analyzing and so on; the discovery of community in biochemistry networks and electronic networks is useful to find the function-related structure units. Finding these communities are not only helps to reveal how the high cohesion and loose coupling community structures are combined, but also helps people to better understand the distinctive characteristics of structures and functions in different levels of the system.

Community detecting can be defined as: finding the vertices with similar characteristics in the complex network and artificially put them into a virtual group. Vertices in the same group have tightly connections while vertices in different groups have sparse connections [7][8]. After division we get a high cohesion and loose coupling system. The feature of community in complex network is similar to the clustering in data mining, so this feature is also called clustering. The ability to find and analyze communities in complex network can provide invaluable help in understanding and visualizing the structure of complex networks [9-20]. In this paper, we show how this can be achieved.

Classical community discovery algorithms (CDA) include, G-N algorithm [10][15], label propagation algorithm [11-13], the CDA based on dynamical similarity[14], to name but a few. With the development of swarm intelligence, some scholars also came up with the idea of CDA based on genetic algorithm (GA) [16-19] and CDA based on particle swarm optimization (PSO)[20-22]. According to the operational solution strategies, the existing CDAs can be divided into two classes, heuristic CDA and optimal CDA. This paper is enlightened both by the thoughts in PSO and by the cooperative thoughts in CF, we entitled it as cooperative and heuristic CDA (CHCDA). Experimental results in section 4 show that the applications can effectively lower the algorithm's time complexity with good practicality and applicability.

II. PRELIMINARIES

With the appearance of Green Computing, it has become a tendency to combine multiple algorithms together to reduce the algorithm's time complexity. Particle swarm algorithm simulates the self-organized sports action in nature; multi-function optimization is able to find multiple optimal swarms at the same time. Therefore, we innovatively apply both these ideas to look

doi:10.4304/jcp.7.1.135-140

for the community seed; then we compare the similarities between free vertices and the existing community. In this paper, we use vertices' degree as the weight of core level, take the core vertex as the best particle, look for and detect the community structures in complex networks.

A. Relation matrix

Assume that the network we studied has n nodes, A is the adjacent matrix and R is the relation matrix.

$$R_n=A_n+E_n \tag{1}$$

From the above formula we can see, the difference between A and R is, the diagonal elements in adjacent matrix is $A_{i,i}=0$ while the relation matrix is $R_{j,j}=1$.

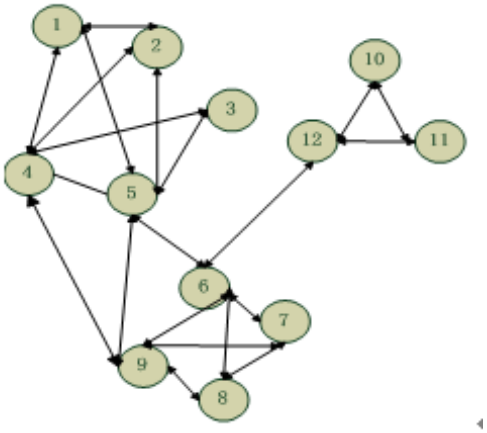


Figure 1. A small artificial network.

We use Figure 1to illustrate how CHCDA works.

The theory of "Six Degrees of Separation" tells us that people who have same friends are much easier to become friends themselves than people who are randomly picked up from the society. Before theorem one, we define the concept of common neighbor and the concept of people with similar social relationships.

Definition 1: Common neighbor. If node a has edges to both node i and j, a is i and j's common neighbor.

Definition 2: Similar social relationship. If i and j have many common neighbors and exceed a minimum threshold, we call i and j have similar social relationships.

In this paper, we use relation similarity to weigh how close the nodes' are. According to the description above we get Theorem 1.

Theorem 1: In network N, We know that A and B are friends, B and C are friends, D and E are randomly chosen from N. We say that A and C are much easier to become friends than D and E. In other words, *similarity(A,C)>>similarity(D,E)*。

Graph theory is always used to solve the problems in network; the problem described above is equal to calculate the probability of triangle formation in an undigraph. In other words, we need to prove that the probability of *Triangle (A,B,C)>>Triangle(D,B,E).*

Proof:

(1) Construct a simple undigraph as Figure 2 shows;
(2) Set node B as a triangle vertex;
(3) If nodes A and C are both B's nearest neighbor, the probability of A, B and C to form a triangle is 50%;
(4) If we randomly choose another two nodes D and E from the graph, the probability of B, D and E to form a triangle is 0.5*0.5*0.5=0.125;
(5) Therefore, the similarity between A and C is much higher than D and E.

Inference 1: Nodes who have similar social relationship are more probable belong to the same community.

We construct the relation matrix in accordance with Figure 1 as below.

$$Matrix=\begin{bmatrix}
1&1&0&1&1&0&0&0&0&0&0&0\\
1&1&0&1&1&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0&0&0\\
1&1&1&1&1&0&0&0&1&0&0&0\\
1&1&1&1&1&1&0&0&1&0&1&0\\
0&0&0&0&1&1&1&1&1&0&0&1\\
0&0&0&0&0&1&1&1&1&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0\\
0&0&0&1&1&1&1&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&1\\
0&0&0&0&0&0&0&0&0&1&1&1\\
0&0&0&0&0&1&0&0&0&1&1&1
\end{bmatrix}$$

By analyzing the matrix we know, the *ith* row of matrix represents the relation vector of i. As a result, we get the degree of i as formula 2 shows.

$$Degree_i=\sum_{j=1}^{n}Matrix_{i,j}-1 \tag{2}$$

By calculating the cosine similarity between relation vectors, we can find the most similar neighbor for each target vertex or community.

B. Community vector

We use vectors to represent the vertices' social relationship. To conveniently calculate the similarity between free vertices and the existing communities, we also use vectors to demonstrate community themes. The

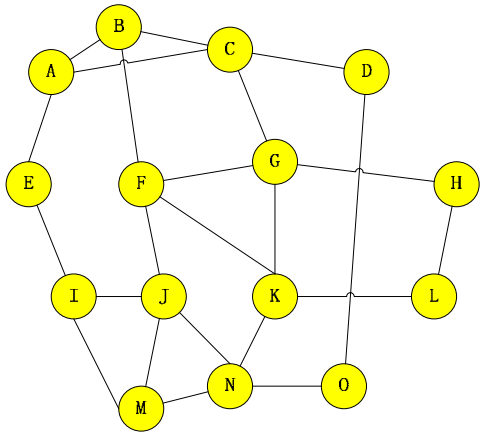


Figure 2. A simple undigraph.

eigenvector of community is combined by the relation vectors of the inner vertices. In this paper, we come up with two methods to demonstrate community eigenvectors.

- Binary vector representation

This method use binary numbers *0* and *1* to represent the community eigenvectors.

For example, the relation matrix of Figure 1 is *Matrix*, the relation vector of vertex *i* is $Matrix_i$. We use V_{SN} to represent the eigenvector of community *SN*, and the vector length is 12, the initial value of V_{SN} is 0. The binary vector representation method can be described as below.

foreach node i in SN

$$V_{SN} = V_{SN} \cup Matrix_i \quad (3)$$

$V_{SN}(i)=1$ means that there is more than one edge to vertex *i*, while $V_{SN}(i)=0$ means none. This method is only able to show whether there is an edge to *i* or not, but not able to count how many. Therefore, we put forward another method to represent the community's relation eigenvector.

- Relation weighted vector representation

This method is able to count the number of edges direct to the target vertex *i*, which to some extent reflects *i*'s attraction. We use $SN_i(a)$ to represent the number of edges in community SN_i to vertex a. The larger of $SN_i(a)$, the bigger of *a's* attraction. The calculation of relation weighted vector can be described as below.

foreach node i in SN

$$V_{SN}=V_{SN}+Matrix_i \quad (4)$$

In this paper, the addition rules are as same as in math. Use Figure 1 as example, vertex 4 and 5 belong to SN_1. If we use binary representation, $V_{SN1}=V_4 \cup V_5=$ *<1,1,1,1,1, 1,0,0,1,0,1,0>*. This result tells us that there are edges direct to vertices *1, 2, 3, 4, 5, 6, 9, 11* in the network, but we do not know how many. If we use relation weighted representation method, $V_{SN1}= V_4+V_5=$*<2,2,2,2,2,1,0,0, 1,0,1,0>*, it tells us that for this moment, there are two edges direct to vertices *1, 2, 3, 4, 5,* and one edge direct to vertices *6, 9, 11*.

Community vector vividly shows the relationship between vertices in the community, which makes it possible to weigh the similarity between free vertices and the existing communities. Besides, it provides an interface for the application of CF to CDA.

C. Community seed

Based on thoughts in multi-function optimization, the best particle leads the rest ones to locate at the most suitable place[23]. Low degree means the vertex locate at the margin place while high degree means it locates at the core position. Community seed is the best local one in one community. We define the concept of community seed and free members as blew.

Definition 3: Community seed. If vertex *a* is the first member of a community, it leads the following members to locate around it. The number one vertex becomes the seed of this community.

Definition 4: Free members. The remained nodes in L_{degree} that haven't been divided into communities are called free members.

The steps to choose community seed are as below.

Step 1: All vertices are sorted in decreasing order of degrees which constitute a list L_{degree}.

Step 2: The set of community seeds *S* is initially set to empty.

Step 3: Vertices in L_{degree} are checked in turn from the beginning to the end of the list. If vertex *i* does not have connection with the existing communities*{SN}*, it becomes a new seed and is added to *S*; if not, calculate the similarity between *i* and *SN* and if similarity*(i, SN)*< δ, *i* become a new seed and be added to *S*; else if similarity*(i, SN)*> δ, *i* becomes a member of *SN*.

Because vertices with higher degrees are checked first, the community seed in each community must be the one with best location. These community seeds respectively guide the rest nodes in the same network to locate at the multiple optima.

The similarity between community *SN* and vertex *i* is calculated as formula (5) shows.

$$Sim(S,i)=cos(S,i)=\frac{V_S \cdot V_i}{|V_S| \cdot |V_i|} \quad (5)$$

When the scale of network changes, it become necessary to change the minimum similarity threshold δ. The larger of the network scale, the more complex of the relationship between vertices. The largest number of edges in a network who have *n* vertices is *(n^2-n)/2*, which means one vertex has a maximum *(n-1)* neighbors. This paper uses relation matrix to represent the relationship between vertices in a network, every vertex has relation to itself which makes the biggest relation number to *n*. The principle of cosine similarity is that the more of the common neighbors that a pair of vertices have, the closer of the relationship between them. In a social network with *n* vertices, the maximum of the common neighbors is *n*, in other words, if two vertices have *n* common neighbors, the similarity between them must be 1, so we set the unit of minimum similarity threshold as *1/n*.

III. CHCDA

In this algorithm, the number of seeds determines the number of communities in the entire network. As a result, we don't need to artificially set the number of communities.

A. Description of community division

We still use Figure 1 to illustrate how this algorithm works out. For Figure 1, $L_{degree}=$*<5,6,4,11,1,2,7, 9,3,8,10,12>*, we set the minimum similarity as 1/12. Vertex *5* has the largest degree, so it becomes the first community seed, we name this community as *A*, and $V_A=V_5$. We check out the free members in L_{degree} one by one, and compare the similarities between them and *A*.

Figure 3 is the dividing result of binary representation and Figure 4 is the dividing result of relation weighted representation.

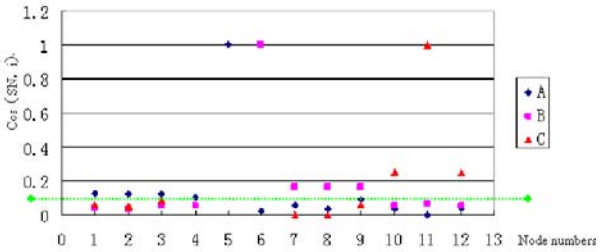


Figure 3. Division of binary community vector representation

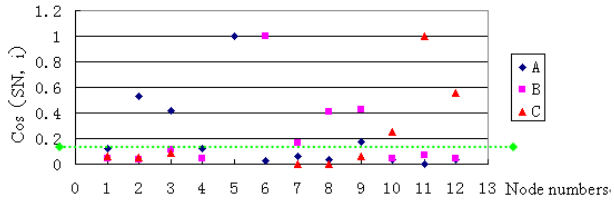


Figure 4. Division of relation weighted community vector representation.

The green dotted line in Figure 3 and Figure 4 is the line of minimum similarity threshold. Compare the results of different community representations, we can see that they get the same community structure, but the relation weighted representation can effectively expand the elasticity of relations between nodes and reduce the fuzzification of node's position.

Besides, there is a distinctive node in figure 4, node *9* belongs to both A and B. However, the similarity between community B and 9 is much higher. We call node 9 the contactor of A and B. Figure 5 shows the dividing results of relation weighted representation.

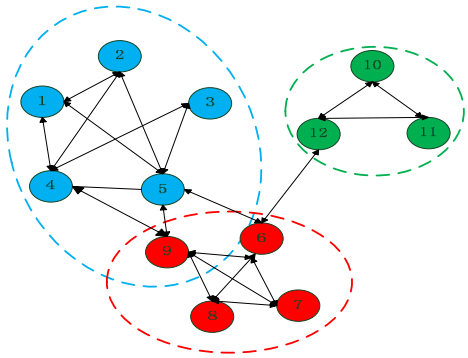


Figure 5. Divided social structure of the network in figure 1.

B. Procedure of cooperative CDA

Step 1: Construct the adjacent map for network N;

Step 2: Build the relation matrix;

Step 3: Sequence the vertices in network N and build up L_{degree};

Step 4: Look for the community seeds, define the community vectors.

Step 5: Divide vertices into different communities.

IV. Experimental Results

, We apply CHCDA to the dividing of Zachary club and dolphins network to test the feasibility of our algorithm.

A. Zachary Club Network

Zachary Club is a one of the most classical network for analyzing network structures. In 1970s, Wayne Zachary took three years to observe the social relationships of members in a US college's karate club, after then he constructed the Zachary's karate club network. The dataset includes 34 nodes and 78 edges, each node represents one member in the club, and edges state the relationship between members. In the procedure of investigation, he found that the director and the coach had disputes with each other about charging problems which result in the whole club break into two parts. The manager and the coach separately became the head of one part. In Figure 6, different colors represent members in different groups. As a real social network, Zachary club is often used to test the efficiency of community discovery algorithm.

Using CHCDA, we firstly construct the relation map of Zachary club. The degree ranking list is L_{degree}=< *34, 1, 33, 3, 2, 4, 32, 9, 24, 6, 7, 8, 14, 15, 28, 30, 31, 5, 11, 20, 25, 26, 29, 10, 13,16,17,18,19,21, 22, 23, 27, 12*>, δ *=1/34*. During the test, we use relation weighted representation to illustrate how this algorithm works well.

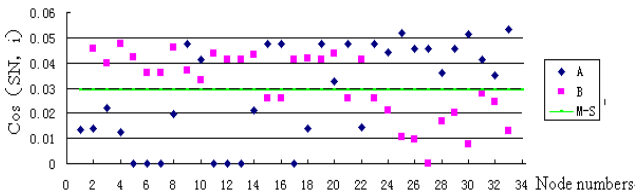


Figure 6. Division results of nodes in Zachary Club

In Figure 6, X axis represents the serial number of nodes in Zachary and Y axis represents the similarity between vertices and the existing communities. As we can see, the similarity values are much smaller than 1, so we take out of the similarity between vertex 1 and community B, vertex 34 and community A because they are too high to mark in the graph. Besides, similarity between 9 and A, 9 and B, 10 and A, 10 and B, 20 and A, 20 and B are all above the minimum similarity threshold, so 9, 10 and 20 are the connectors for community A and B, Which shows that CHCDA is really effective for discovering the overlapped area of communities. Figure 7 is the dividing result of CHCDA.

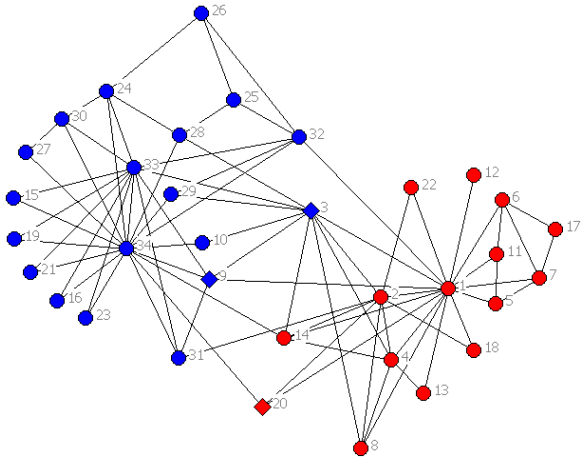


Figure 7. CHCDA division results for Zachary Club

B. Dolphins Network

Dolphin network is also a common dataset in the study of social networks. Lusseau and his fellows conduct systematic surveys about dolphins in Doubtful Sound, Fiordland, New Zealand. The survey course constantly remains over 7 years and covers the entire home range of Doubtful Sound population which includes 62 vertices and 159 edges. In Figure 8, each vertex represents a dolphin; a link means two dolphins have regular contact. We set the minimum similarity as 1/62 and get Figure 9.

Figure 8 shows that the entire network is divided into two parts. Besides, there are some special dolphins locate at the overlapped area, such as vertex 20, 31 and 37.

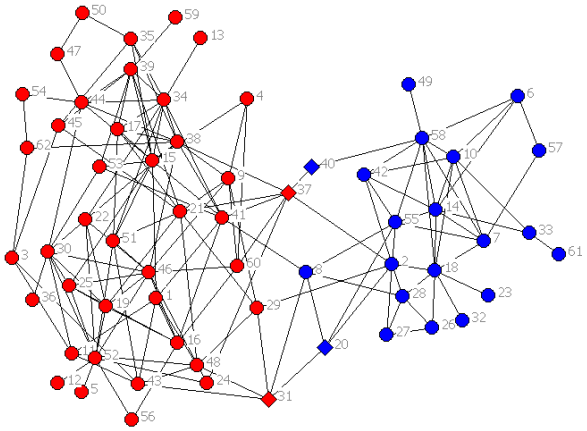


Figure 8. CHCDA division results for Dolphin

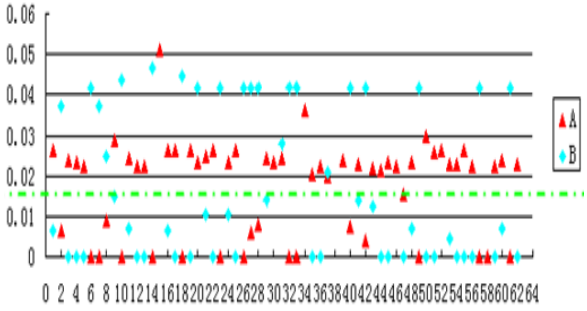


Figure 9. Division results of nodes in Dolphin

Individuals who lie on the boundaries of communities become bridge between unconnected communities. They play important roles in the procedure of information spreading. The discovery of connectors has great value to promote the communication and interaction between communities, especially in the E-commence and citation network.

V. Conclusion

In this paper, we have analyzed the problem of detecting community structure in networks. During the research, we come up with a cooperative community detecting algorithm based on the ideas in multi-model optimization. We have three innovation points in this paper: (1) Introduce the concept of community seed, declare that community seeds guide the rest individuals to locate at the multiple optima. (2) Put over the idea of relation matrix, use relation similarity to weigh the social intimacy between nodes. (3) Apply CF's thoughts to community discovery; find the contactors between different communities. Those three characteristics enable the CHCDA algorithm efficiently finds the potential communities in large complex network.

Follow-up works include formulate the minimum similarity threshold and choose suited community vector representation for different social networks.

References

[1] M. E. J. Newman. "Finding community Structure in Networks Using the Eigenvectors of Matrices." Physics Review E. 2006, 1–22.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex Networks: Structure and Dynamics." Physics Reports 424, 2006, 175–308.

[3] Watts D J, Strogatz S H. "Collective Dynamics of Small-World' Networks." Nature, 1998, 393(6638): 440-442.

[4] Duan Xiao-dong, Wang Cun-rui, Liu Xiang-dong. "Web Community Detection Model Using Particle Swarm Optimization." Computer Science, 2008, 35(3): 18-22.

[5] Albert R, Jeong H, Barabasi AL. "The Internet's Achilles Heel: Error and Attack Tolerance of Complex Networks." Nature, 2000, 406(2115): 378-382.

[6] Barabasi AL, Albert R. "Emergence of Scaling in Random Networks." Science, 1999, 286(5439):509-512.

[7] Guimera` R, Amaral L. A. N. "Functional Cartography of Complex Metabolic Networks." Nature, 2005, 433(7028): 895-900.

[8] Palla G, Der′ enyi I, Farkas I,Vicsek T. "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society." Nature, 2005, 435(7043): 814-818.

[9] HE Dong-Xiao, ZHOU Xu, WANG Zuo, ZHOU Chun-Guang, WANG Zhe, JIN Di. "Community Mining in Complex Networks—Cluster Genetic Algorithm." ACTA AUTOMATICA SINICA, 2010, 36(8): 1160-1170.

[10] Girvan M,Newman M E J. "Community Structure in Social and Biological Networks". Proceedings of the National Academy of Sciences of the United States of America, 2002, 99(12): 7821-7826

[11] Raghavan U N, Albert R,Kumara S. "Near Linear Time Algorithm to Detect Community Structures in Large-scale Networks." Physical Review E, 2007, 76(3): 036106.

[12] Barber M J,Clark J W. “Detecting Network Communities by Propagating Labels Under Constraints.” Physical Review E, 2009, 80(2): 026129.
[13] Leung I X Y,Hui P,Lio` P,Crowcroft J. “Towards Real-time Community Detection in Large Networks.” Physical Review E,2009,79(6):066107.
[14] Zhang Y Z, Wang J Y, Wang Y,Zhou L Z. “Parallel Community Detection on Large Networks with Propinquity Dynamics.” In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France: ACM, 2009, 997-1006.
[15] Newman M E J. “Fast Algorithm for Detecting Community Structure in Networks.” Physical Review E,2004, 69(6): 066133.
[16] Liu X, Li D Y, Wang S L, Tao Z W. “Effective Algorithm for Detecting Community Structure in Complex Networks Based on GA and Clustering. “ In: Proceedings of the 7rded International Conference on Computational Science. Beijing, China， Springer, 2007. 657-664.
[17] Gog A, Dumitrescu D, Hirsbrunner B. “Community Detection in Complex Networks Using Collaborative Evolutionary Algorithms.” In: Proceedings of the 9thed European Conference on Artificial Life. Lisbon, Portugal: Springer, 2007. 886-894.
[18] Tasgin M, Herdagdelen A, Bingol H. “Community Detection in Complex Networks Using Genetic Algorithms.” http://arxiv.org/abs/0711.0491, 2010.
[19] Pizzuti C. “A Genetic Algorithm for Community Detection in Social Networks.” In: Proceedings of the 10thed International Conference on Parallel Problem Solving from Nature. Dortmund, Germany, Springer, 2008. 1081-1090.
[20] DAI Fei-fei, TANG Pu-ying. “Community Structure Detection in Complex Networks Using Particle Swarm Optimization Algorithm.” Computer Engineering and Applications, 2008, 44(22): 56-58.
[21] GAO Chun-tao. “Research on Particle Swarm Optimization and Its Application.” Journal of Harbin University of Commerce (Natural Sciences Edition). 2010, 26(4): 442-445.
[22] X. Li. “Adaptively Choosing Neighborhood Bests using Species in A Particle Swarm Optimizer for Multimodal Function Optimization,” Proceedings of Genetic and Evolutionary Computation Conference, 2004, 105-116.
[23] Li, J.P., Balazs, M.E., Parks, G. and Clarkson, P.J. “A Species Conserving Genetic Algorithm for Multimodal Function Optimization.” Evolutionary Computation. 2002, 10(3): 207–234.

Ruixin Ma, (1975--). Lecturer of Dalian University of Technology. Research area: E-commence, community discovery and swarm intelligence.

Guishi Deng, (1945--). Professor of Dalian University of Technology. Research area: E-commence, decision analysis and analysis of complex system.

Wang Xiao, (1988--). Ph.D. candidate at Institute of Automation, Chinese Academy of Science. Research area: E-commence, community discovery and swarm intelligence.

Workload-aware Reliability Evaluation Model in Grid Computing

Peng Xiao
School of Computer and Communication, Hunnan Institute of Engineering, Xiangtan, China
Email: xpeng4623@yahoo.com.cn

Zhigang Hu
School of Information Science and Technology, Central South University, Changsha, China
Email: zghu@mail.csu.edu.cn

***Abstract*—To address the issue of reliability evaluation in dynamic grid environment, a workload-aware reliability evaluation model is proposed, in which queuing system is applied to describe the dynamic workload and working of grid resources. Meanwhile, a new type of resource fault, namely Deadline-Miss fault, is introduced to evaluate the reliability of real-time jobs. The validity of the proposed model and its approach to calculate real-time job's reliability are presented theoretically. Extensive experiments are conducted to verify its performance, and the results show that the proposed model can significantly improve the accuracy of reliability evaluation in presence of dynamic workload. Also, scheduling algorithm based on this model can reduce the jobs' mean response time and the deadline-miss rate.**

***Index Terms*—grid computing, reliability evaluation, workload-aware, deadline-miss, queuing system**

I. Introduction

Grid computing [1] is an emerging technology to address issues such as large-scale resource sharing, wide-area communication, and multi-institutional collaboration. Due to its open architecture, grid resources from different virtual organizations are inherently heterogeneous and dynamic, which lead to resource failures occur frequently and hard to be detected [2, 3]. Furthermore, wide-area resource sharing over uncertain network increases the probability of resource failures since data transferring are often intensive in most of grid applications [4]. Therefore, reliability analysis and evaluation of grid systems becomes a challenging issue and attracts more and more attentions of researchers. Existing studies show that the difficulties of reliability evaluation in grid environments are as following:

- Heterogeneity of resources results in various types of resource fault, which can hardly be taken into consideration in a single model [3, 19, 20].
- Failure model is greatly influenced by resource's workload, which often fluctuates dramatically and unpredictable in runtime [2-3].
- Due to the large-scale and the complexity of grid systems, reliability are difficult to model, analyze, and evaluate [2-5].

Among the existing grid reliability evaluation models, the *Tree-structured Grid Reliability Model* [2] (TGRM) proposed by Y.S. Dai *et al*. is considered as the most effective and efficient. However, the shortcoming of TGRM is that it does not take into account the dynamic workload when calculating reliability. Motivated by this observation, in this paper we propose a workload-aware reliability evaluation model which is based on TGRM.

The rest of this paper is organized as follows: Section 2 presents the related work. In Section 3, we analyze the shortcomings of TGRM, and present the workload-aware TGRM. In Section 4, extensive simulations are conducted to verify the performance of the proposed model. Finally, Section 5 concludes the paper with a brief discussion of future work.

II. Related work

In [6], *Distributed Program Reliability* (DPR) and *Distributed System Reliability* (DSR) are two metrics firstly introduced to the reliability evaluation of distributed systems by Raghavendra. Also, Raghavendra propose an algorithm based on graph-traversal to calculate DSR and DPR. However, the complexity of the algorithm is exponential, which means it is unsuitable to evaluation the reliability of large-scale systems. So, Chen *et al.* [7] improves Raghavendra's algorithm by using the graph-cutting approach. The common shortcoming of both two algorithms is that they all assume that the failure rates of resources and links are constant, which is unsuitable to the geographically distributed and dynamic grid environments.

So, in [5] Y.S. Dai *et al.* addresses the issue of reliability evaluation in grid environments. Dai's approach is based on *Minimal-Task-Spanning-Tree* (MTST) just like Raghavendra's, but he uses random failure model instead of assuming the failure rate being constant. In [2], Y.S. Dai *et al.* further their study and propose a *Tree-structured Grid Reliability Model* (TGRM) for analyzing the performance and the reliability of grid services in the presence of common failures caused by sharing communication links. TGRM is the first grid reliability model and has been proven to be effective and efficient.

doi:10.4304/jcp.7.1.141-146

However, TGRM dose not take into account the effects of workload on the reliability evaluation. In [3], Czajkowski has pointed out that "The reliability of grid system is greatly affected by many factors, and the dynamic changing workload on resources is one of the most significant one". Many other studies also prove this conclusion. For example, in [8] Kermarrec *at el.* gives the formal expression to describe the relationship between performance and workload in large-scale distributed systems. In [9-10], Bucur and Epema conduct extensive experiments in grid testbed DAS-2 [11] to evaluate the performance of various co-allocation policies under different workload. Their experimental results show that workload-aware co-allocation policies are effective to reduce the probabilities of executing fault, since those policies are effective to avoid the negative effects brought by dynamic changing of workload on resources. In addition, the studies in [5, 7, 10, 22, 23] confirm that the conception of workload-aware is of significant importance not only for the performance of scheduler but also for the whole performance of grid systems.

In this paper, we apply queuing system [15] to describe the workload of grid resources. Recently, queuing system has been widely applied by many researchers to model the working of grid resources. For example, Sun Xian-He [12] has used queuing theory to predicate the availability of grid resources; Wu Ming [13] has applied M/G/1 queuing system to describe working model of single resource to analyze the effects of advance reservation on local job's scheduling; Berten [14] uses M/M/C queuing system to model the service of a single cluster to study the capability-based allocation policy in multi-cluster grid. Their studies indict that queuing system is capable of precisely describing the working and workload model of grid resources.

III. Reliability model analysis and design

A. TGRM Analysis

In TGRM, RMS (Resource Management System) is the root node of the tree model, which is responsible for task admission and scheduling. The resources are represented by leaf nodes, between which the edges represent the network links. TGRM assumes that the occurrence of resource and link faults follow exponential distribution. In order to simplify the evaluation of reliability, TGRM make three assumptions as following:

A1. The RMS is completely reliable, which means there will be no faults in RMS.

A2. The RMS receives tasks, assigns them to available resource, and integrates the received results for user's tasks. All the intermediate data are assembled on RMS, and then being sent back to users.

A3. Each available resource can only provide service for a single user's task.

Based on the above assumptions, TGRM gives the expressions to calculating the reliabilities of the resources and the links as following:

$$RF_j^k = e^{-\varpi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k})} \quad (1)$$

$$LF_j^k = e^{-\pi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k})} \quad (2)$$

where RF_j^k is the probability that sub-task j is successful completed on resource k, LF_j^k is the probability that there is no occurrence of link fault during the execution of sub-task j, ϖ_k and π_k are the mean fault rates of the resource k and the links, c_j is the size of sub-task j, a_j is the amount of data transmitted between resource k and RMS when sub-task j is running, x_k is the computing speed of resource k, s_k the minimal bandwidth of all the network links between resource k and RMS. Therefore, the reliability of a task is that

$$\text{Reliability} = \min_{1 \le j \le m} [\max_{k \in w_j} (RF_j^k \cdot LF_j^k)] \quad (3)$$

where m is the number of sub-tasks, and w_j is the set of selected resources for executing the task.

B. Workload-aware TGRM

As mentioned in Section IV, queuing system has been proven to be able to precisely describe the working and workload model of grid resources. So, we also use queuing system to construct the working and workload model of grid resources. By this way, we incorporate workload-aware mechanism into the TGRM.

Among the assumptions of TGRM, it is obviously that the third assumption is too conservative and unsuitable. For instance, multi-cluster systems are a typical class of grid system that generally used in scientific fields. In multi-cluster systems, a high-performance cluster exposes its service interface to the out-side users in form of *single-image*. So, a cluster should be considered as a single resource with paralleling computing capability. This is also true for those *Massive Parallel Processor* (MPP) systems. To be compatible with TGRM, we use M/M/1 queuing system to model those resources that can only provide service for a single task at a time, and M/M/C queuing system for those resources with parallel capability.

Meanwhile, as shown in (1), (2) and (3), TGRM ignores the waiting time of jobs when calculating the reliability. It may be feasible when resources are abundant or workload is in low-level, but not suitable to high-level workload which is common in grid environments. Another problem of TGRM is that it does not take into consideration the deadline constraint, which is often required by real-time tasks. For those real-time tasks, if it cannot be completed before the deadline the results of the tasks are of less or no value for users. So, deadline-miss should be considered a sort of resource failure.

To overcoming the above shortcomings of TGRM, we relax the third assumption of TGRM to provide supports for parallel serving resources. Also, we introduce a new type of fault, called *Deadline-Miss Fault* (DMF), into TGRM to support the reliability evaluation of real-time tasks. Therefore, equation (1), (2) and (3) listed previously is modified as following:

$$RF_j^k = e^{-\varpi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k} + d_j)} \quad (4)$$

$$LF_j^k = e^{-\pi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k} + d_j)} \quad (5)$$

$$\text{Reliability} = \min_{1 \le j \le m} [\max_{k \in w_j} (RF_j^k \cdot LF_j^k \cdot DMF_j^k)] \quad (6)$$

where d_j is the deadline requirement of sub-task j, DMF_j^k is the probability that there is no deadline-miss when sub-task j is scheduled on resource k.

As shown in (6), the key issue of workload-aware TGRM is the approach to calculate the DMF_j^k, which is greatly affected by the workload on resources. So, in the next section, we will focus on the calculation of DMF_j^k by using queuing system.

C. Calculation of Deadline Miss Failure

As mentioned above, workload-aware TGRM uses M/M/1 queuing system to model those resources that can only provide service for a single task at a time, and M/M/C queuing system for those resources with parallel capability (i.e. cluster or MPP). Therefore, the approaches to calculate the DMF_j^k for these two types of resources are different.

Theorem 1. If resource R_k is modeled as M/M/C_k queuing system, then the probability that there is no deadline-miss when sub-task j is scheduled on R_k is

$$DMF_j^k = \sum_{n=0}^{C_k} \delta \cdot \frac{(\rho_k \cdot C_k)^n}{n!} + \sum_{n=1}^{C_k \cdot \mu_k \cdot d_j - 1} \delta \cdot \frac{\rho_i^{n+C_k} \cdot C_k^{C_k}}{C_k!}$$

where $\delta = \left[\sum_{n=1}^{C_k} \frac{(\rho_k \cdot C_k)^n}{n!} + \frac{(\rho_k \cdot C_k)^{C_k}}{C_k} \frac{1}{1-\rho_k} \right]^{-1}$, $\rho_k = \lambda_k / (C_k \cdot \mu_k)$, d_j is the deadline of sub-task j, λ_k is the mean interval of sub-tasks on R_k, μ_k the mean service time of R_k, and C_k is the paralleling processing capability of R_k.

Proof. Let ψ be the random variable representing the number of waiting tasks in R_k. According to queuing theory [15], the probability that there are m waiting jobs in R_k is

$$\Pr\{\psi = m\} = \begin{cases} \delta \cdot \frac{\rho_k^{m+C_k} \cdot C_k^{C_k}}{C_k!}, & m > 0 \\ \sum_{n=0}^{C_k} \delta \cdot \frac{(\rho_k \cdot C_k)^n}{n!}, & m = 0 \end{cases} \quad (7)$$

$$\text{where } \delta = \left[\sum_{n=1}^{c_i} \frac{(\rho_i \cdot c_i)^n}{n!} + \frac{(\rho_i \cdot c_i)^{c_i}}{c_i} \frac{1}{1-\rho_i} \right]^{-1} \quad (8)$$

For M/M/C_k queuing system, the service rate is $C_k \cdot \mu_k$, which means the system can complete $C_k \cdot \mu_k$ tasks in a unit time. So, the amount of tasks that R_k can complete in period d_j is $C_k \cdot \mu_k \cdot d_j$. Therefore, the probability that R_k can guarantee a tasks's deadline d_j is equal to the probability that the waiting tasks in R_k is not more than $C_k \cdot \mu_k \cdot d_j - 1$. That is

$$DMF_j^k = \Pr\{\psi \le c_i \cdot \mu_i \cdot d_j - 1\} \quad (9)$$

Combing formulas (7)(8)(9) and the above analysis, we can get that

$$\begin{aligned} DMF_j^k &= \Pr\{\psi \le C_k \cdot \mu_k \cdot d_j - 1\} \\ &= \sum_{m=0}^{C_k \cdot \mu_k \cdot d_j - 1} \Pr\{\psi = m\} \\ &= \sum_{m=0}^{C_k} \delta \cdot \frac{(\rho_k \cdot C_k)^m}{m!} + \sum_{m=1}^{C_k \cdot \mu_k \cdot d_j - 1} \delta \cdot \frac{\rho_k^{m+C_k} \cdot C_k^{C_k}}{C_k!} \end{aligned} \quad (10)$$

■

Theorem 2. If resource R_k is modeled as M/M/1queueing system, then the probability that there is no deadline-miss when task j is scheduled on R_k is

$$DMF_j^k = \sum_{m=0}^{\mu_k \cdot d_j - 1} (1-\rho_k) \cdot \rho_k^m$$

where $\rho_k = \lambda_k / \mu_k$, d_j is the deadline of task j, λ_k is the mean interval of sub-tasks on R_k, μ_k the mean service time of R_k.

Proof. Let ψ be the random variable representing the number of waiting tasks in R_k. According to queuing theory [15], the probability that there are m waiting jobs in R_k is

$$\Pr\{\psi = m\} = (1-\rho_k) \cdot \rho_k^m, \quad m \ge 0 \quad (11)$$

For M/M/C_k queuing system, the service rate is μ_k, which means the system can complete μ_k tasks in a unit time. Similar to the analysis in Theorem 1, we can obtain that

$$\begin{aligned} DMF_j^k &= \Pr\{\psi \le \mu_i \cdot d_j - 1\} \\ &= \sum_{m=0}^{\mu_k \cdot d_j - 1} \Pr\{\psi = m\} \\ &= \sum_{m=0}^{\mu_k \cdot d_j - 1} (1-\rho_k) \cdot \rho_k^m \end{aligned} \quad (12)$$

■

Based on Theorem 1 and Theorem 2, we can obtain the workload-aware TGRM by substitute the (10) or (12) into in (4), (5) and (6).

IV. Experiments and Performance Comparison

A. Experimental Settings

In experiments, we use GridSim [16], a distributed resource management and scheduling simulator, to construct a multi-cluster grid model. The grid model follows the TGRM's tree-structured, and its topology and setting of individual resources are deprived from the grid test-bed DAS-2 [11]. As shown in Fig. 1, the grid model consists of twelve *Computational Elements* (CE 1 ~ CE 12), each representing a high-performance cluster. The clusters are grouped into five groups by their geographical positions. Within each group, the clusters are connected by LAN (Link 1 ~ Link 12). Then, they are connected by WAN (Link 13 ~ Link 17) between groups. The failure of links follows exponential distribution with various parameters. According to the statistics in [17, 19, 21], the failures of LAN link are different from the

WAN's. So, we set that the LAN's fault rate limited in $[0.02, 0.04]$, and the WAN's fault rate limited in $[0.04, 0.12]$.

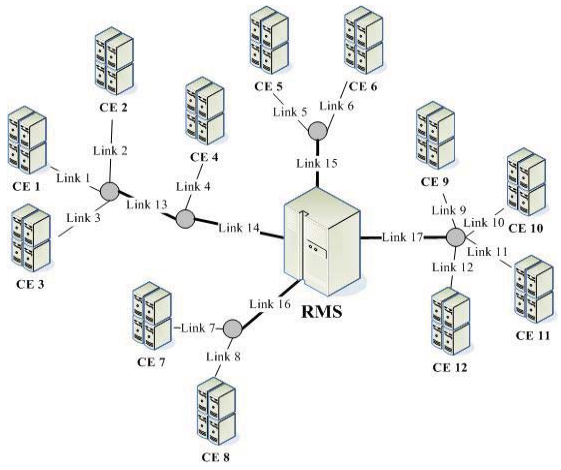


Figure 1. Grid Model in Simulations

The detailed configurations of each CE are presented in Table 1, which is also deprived from grid test-bed DAS-2.

TABLE 1
Setting of Computational Grid Model

ID	Processor number	MIPS / processor
CE 1	64	377
CE 2	128	410
CE 3	256	380
CE 4	128	285
CE 5	128	285
CE_6	64	515
CE 7	128	215
CE 8	64	285
CE 9	256	380
CE 10	512	215
CE 11	256	333
CE 12	128	410

In simulations, the basic workload (tasks stream) is generated by using Lublin-Feitelson model [18], which is derived from the logs of real supercomputers. It consists of 10000 tasks, each is characterized by its arrival time T_a, resource demands R, execution time T_e. However, this basic workload can not meet the requirements of our simulation, because it lacks of deadline and time of data transmission. So, we modify the basic workload by append each task with deadline d and data transmission time T_{data}, which is obtained as following.

$$d = T_a + g \cdot T_e \tag{13}$$

$$T_{data} = h \cdot T_e \tag{14}$$

where g is a random variable that uniformly distributed in $[5.5,10.5]$, and h is a uniformly distributed in $[0.5,1.5]$. Therefore, each task in the modified workload is characterized by five-tuple: $< T_a, T_e, T_{data}, R, d >$.

B. Comparison of Accuracy

In simulations, we mainly focus on the accuracy of *Workload-aware TGRM* (WA_TGRM) comparing with TGRM. Also, we investigate WA_TGRM's performance in term of *Mean Response Time* (MRT). As mentioned in Section 3.2, the differences between WA_TGRM and TGRM are the reliability calculating formulas as shown in (4), (5) and (6). So, the algorithms of reliability calculation and task scheduling in WA_TGRM are as same as that in TGRM, which are specified in the appendix of [2].

In this experiment, we first calculate the reliabilities of tasks by using TGRM and WA_TGRM respectively, then schedule the tasks onto resources. As to each task, there are three type of executing results: success, abort (resource fault or link fault), deadline-miss fault. The experimental results are shown in Fig. 2, in which two curves represent the reliabilities calculated by TGRM and WA_TGRM. The third curve is the real reliability that calculated as $\frac{amount\ of\ successful\ tasks}{amount\ of\ total\ tasks}$.

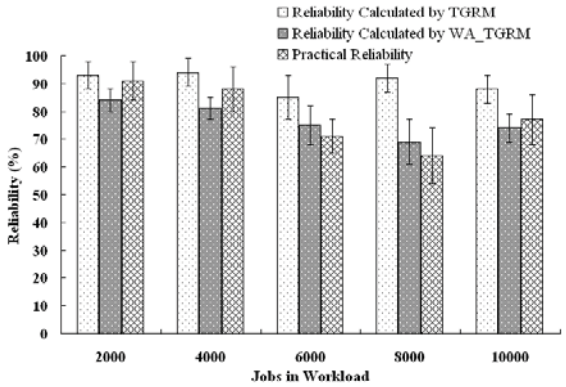


Fig. 2 Accuracy of TGRM and WA_TGRM

We define the ratio of theoretic reliability to real reliability as the accuracy of TGRM or WA_TGRM. As shown in Fig. 2, the accuracy of TGRM is about 97% for the first 2000 tasks, and 93% for the second 2000 tasks. As to WA_TGRM, they are about 92% and 91% correspondingly. However, the high accuracy of TGRM does not be maintained with the continuing of the simulation. In fact, TGRM's accuracy decreases dramatically for the third and the fourth 2000 tasks, which are only about 84% and 69%. On the contrary, WA_TGRM's accuracy is always kept above 90%. In order to examine such results, we record all the causes of unsuccessful execution and their percentages, which are shown in Fig. 3. As we can see, the percentages of resource fault and link fault are very high for the first 4000 tasks. However, percentage of deadline-miss fault keeps increasing with the continuing of the experiment, which becomes the highest (about 34%) when executing the 6000^{th} - 8000^{th} tasks.

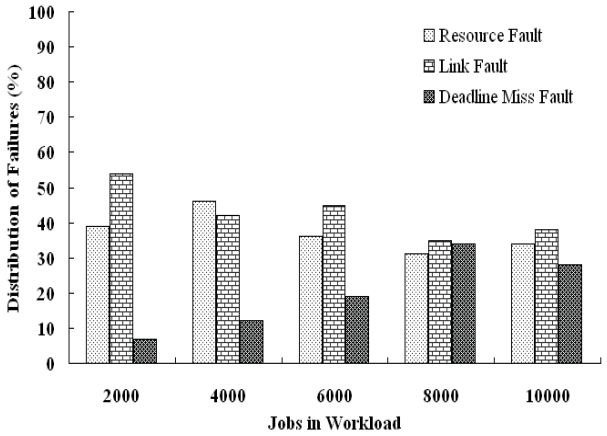


Fig. 3 Distribution of Execution Failure Causes

The detailed observation of the simulation shows that most of the tasks are scheduled onto low-load resources at the beginning. With the increasing of new arriving tasks, the workload of resources gradually becomes heavier and heavier. As a result, the waiting time of tasks increased consequently, which causes higher deadline-miss fault. As TGRM does not take the waiting time into account, nor dose it consider deadline-miss fault as a type of fault, its accuracy inevitably decreases in presence of high deadline-miss fault just as shown in Fig. 2. On the contrary, WA_TGRM overcomes those TGRM's shortcomings. As a result, TGRM's accuracy can always be kept in high-level in various cases, which indicates that WA_TGRM is more adaptive than TGRM in practice grid systems.

C. Comparison of Scheduling Performance

We investigate the scheduling performance of TGRM and WA_TGRM in term of *Mean Response Time* (MRT). Although the scheduling algorithm of both TGRM and WA_TGRM are the same, their scheduling schemes (*Minimal-Task-Spanning-Tree, MTST)* for individual tasks are different, because their reliability evaluation formulas are different. The experimental results are show in Fig. 4.

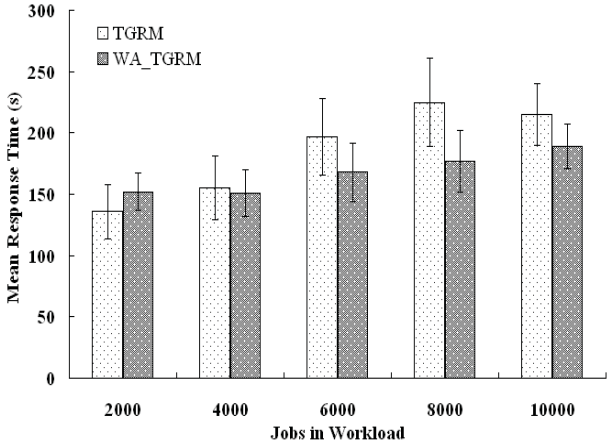


Fig. 4 Mean Response Time of Jobs

The results show that the mean response time of WA_TGRM is about 20% lower than TGRM since after executing the first 4000 tasks. The detailed observation of the simulation shows that the MTSTs generated by WA_TGRM often obtains many low-load resources. As shown in (6), DMF_j^k is an important factor when calculating reliability in WA_TGRM. The effects of DMF_j^k on the results of reliability evaluation are influenced by the resource's workload, which can be clearly seen from the conclusions in Theorem 1 and Theorem 2. More specifically, when workload is low the effects of DMF_j^k is also lower, and vice versa. So, when the system is in face of the heaviest workload (during the 6000^{th} – 8000^{th} tasks) WA_TGRM is more effective to be aware such heavy workload and generate better scheduling scheme. The experimental results further confirm the Bucur's conclusion that "workload-aware scheduling algorithm is more effective to reduce mean response time and load-balance" in [9].

V. CONCLUSION

In this paper, we propose a novel Workload-aware Reliability evaluation model, which is deprived from TGRM. In the proposed model, queuing system is applied to describe the workload grid resources. In order to evaluate the reliability of jobs with constrain to deadline, a new type of resource fault (Deadline-Miss fault) is introduced to the proposed model. In the simulations, we compare the performance of WA_TGRM with TGRM in terms of accuracy. The experimental results show that WA_TGRM can provide more accurate reliability evaluation when grid system in presence of high-level workload. This indicates that WA_TGRM is more adaptive than TGRM in practice grid systems. Also, simulations are conducted to examine the performance of reliability-based job scheduling. Experimental results show that WA_TGRM is able to reduce the jobs' mean response time about 15% because its ability to generate workload-aware scheduling schemes.

At present, our WA_TGRM concentrates on the reliability evaluation when allocating resources. In practice grid systems, advance reservation is an effective technique that generally used to improve the reliability of co-allocation from multi-institutions. So, our future work is to take advance reservation into account when calculating resources' reliability. More specific, we plan to introduce another type of fault, namely Reservation-miss Fault, into current WA_TGRM design.

ACKNOWLEDGMENT

This work was supported by a grant from the National Natural Science Foundation of China (No. 60673165 and No. 60970038).

REFERENCES

[1] I. Foster, C. Kesselman. The Grid2: Blueprint for a New Computing Infrastructure. San Francisco: Morgan Kaufmann, 2004.

[2] Y.S. Dai, G. Levitin. Reliability and Performance of Tree-Structured Grid Services. IEEE Transactions on Reliability, 2006, 55(2):337-349.
[3] K. Czajkowski, I. Foster, C. Kesselman. Resource Co-Allocation in Computational Grids. Proceedings of Inte'l Symp. on High Performance Distributed Computing, 1999:219-228.
[4] Y.W. Li, Z.L. Lan. Exploit Failure Prediction for Adaptive Fault-Tolerance in Cluster. Proceedings of IEEE Inte'l Symp. on Cluster Computing and the Grid, 2006:531-538.
[5] Y.S. Dai, M. Xie, K.L. Poh. Reliability Analysis of Grid Computing Systems. Proceedings of Pacific Rim Inte'l Symp. on Dependable Computing, 2002:97-104.
[6] C.S. Raghavendra, V.K.P. Kumar, S. Hariri. Reliability Analysis in Distributed Systems. IEEE Transactions on Computers, 1988, 37(3):352-358.
[7] D.J. Chen, T.H. Huang. Reliability Analysis of Distributed Systems Based on a Fast Reliability Algorithm. IEEE Transactions on Parallel and Distributed Systems, 1992, 3(2):139-154.
[8] A.M. Kermarrec, L. Massoulie, A.J. Ganesh. Probabilistic Reliable Dissemination in Large-Scale Systems. IEEE Transactions on Parallel and Distributed System, 2003, 14(3):248-258.
[9] A.I.D. Bucur, D.H.J. Epema. Scheduling Policies for Processor Coallocation in Multicluster System. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(7):958-962.
[10] A.I.D. Bucur, D.H.J. Epema. The Performance of Processor Co-Allocation in Multicluster Systems. Proceedings of IEEE Inte'l Symp. on Cluster Computing and the Grid, 2003:302-309.
[11] H. Bal, R. R. Bhoedjang, R. Hofman, et al. The Distributed ASCI Supercomputer Project. ACM Operating Systems Review, 2000, 34(4):76-96.
[12] X.H. Sun, M. Wu. Grid Harvest Service: A System for Long-Term, Application-Level Task Scheduling. Proceedings of Inte'l Symp. on Parallel and Distributed Processing (IPDPS'03), 2003.
[13] M. Wu, X.H. Sun, Y. Chen. QoS Oriented Resource Reservation in Shared Environments. Proceedings of Inte'l Symp. on Cluster Computing and the Grid, 2006:601-608.
[14] V. Berten, J. Goossens, E. Jeannot. On the Distribution of Sequential Jobs in Random Brokering for Heterogeneous Computational Grids. IEEE Transactions on Parallel and Distributed Systems, 2006, 17(2):113-124.
[15] D. Gross, C.M. Harris. Fundamentals of Queuing Theory. USA: John Wiley and Sons, 1998.
[16] R. Buyya, M. Murshed. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. Journal of Concurrency and Computation: Practice & Experience, 2002, 14:1175-1220.
[17] D.M. Nicol, W.H. Sanders, K.S. Trivedi. Model-Based Evaluation: From Dependability to Security. IEEE Transactions on Dependable and Secure Computing, 2004, 1(1):48-65.
[18] U. Lublin, D.G. Feitelson. The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs. Journal of Parallel and Distributed Computing, 2003, 63(11):1105-1122.
[19] D.K. Kim, Y. Jiao, E. Tilevich. Flexible and Efficient In-Vivo Enhancement for Grid Applications. Proceedings of International Symp. on Cluster Computing and the Grid, 2009:444-451.
[20] H. Zhao, X.L. Li. Efficient Grid Task-Bundle Allocation Using Bargaining Based Self-Adaptive Auction. Proceedings of International Symp. on Cluster Computing and the Grid, 2009:4-11.
[21] M. Waldburger, M. Göhner, H. Reiser, G.D. Rodosek, B. Stiller. Evaluation of an Accounting Model for Dynamic Virtual Organizations. Journal of Grid Computing, 2009, 7(2):181–204.
[22] D. Lingrand, J. Montagnat, T. Glatard. Modeling the Latency on Production Grids with Respect to the Execution Context. Proceedings of Inte'l Symp. on Cluster Computing and the Grid, 2008:753-758.
[23] K Verstoep, J Maassen, H E Bal, J W Romein. Experiences with Fine-grained Distributed Supercomputing on a 10G Testbed. Proceedings of Inte'l Symp. on Cluster Computing and the Grid, 2008::376-383.

Peng Xiao was born in 1979. He received his master degree in Xiamen Universy in 2004. Now, he is a Ph.D candidate in Central South University. His research interests include grid computing, parallel and distributed systems, network computing, distributed intelligence.

Zhigang Hu was born 1963. He is a Professor and Ph.D supervisor in Central South University. Currently, he is the Chief Secretary of Computer Science in Hunan Province. His research interests including grid computing, embedded systems, high-performance platform.

Battery and Power Consumption of Pocket PCs

Assim Sagahyroon
Department of Computer Science and Engineering
American University of Sharjah, UAE
Email:asagahyroon@aus.edu

***Abstract*—Due to the increased functionality in today's portable devices, battery life and energy consumption continue to be a major concern for both designers and users. Unfortunately battery technology is not keeping pace with the energy requirements of these devices and therefore energy-efficient hardware design techniques, software optimization, energy management, and the design of efficient communication protocols continue to be explored by researches as viable means that would assist in the efficient use of the energy resources of these mobile devices. In this paper, using results obtained from a number of experiments, we investigate battery utilization and power consumption of two pocket PCs. Using benchmarks that are representative of typical workloads, and by varying the operating conditions of these devices we were able to explore the impact of a number of features available in these devices on battery life and the power consumed. Characterizing the consumption of these devices provides a platform for further research and contribute to the design of improved and energy-efficient future mobile computing devices.**

***Index Terms* —portable devices, battery life, power consumption.**

I. INTRODUCTION

Mobile devices are increasingly becoming an integral part of our daily life contributing to an ever growing ubiquitous computing environment. Hence, users will continue to expect more functionality from these devices. Enhanced features and the complex operations performed by these portable devices are rapidly draining their limited system energy budget. Battery lifetime is perhaps one of the most important characteristics of a portable computer. For many users, doubling the battery lifetime may be far more important than doubling the clock frequency. A main concern is the fact that improvements in battery capacity have not kept pace with the improvements in microelectronics technology [1]. Reducing power consumption and prolonging battery-life is a major concern for designers. Battery capacity grows very slowly, about 5 to 10% every year, which is not adequate when we take into consideration the energy demand of handheld devices [2]. An understanding of the power consumption behavior of these devices and the impact of the different operating modes and scenarios on battery life is critical for the design of energy efficient devices.

In [3], an attempt was made to study the power usage of an IBM ThinkPad R40 laptop. The authors run a set of experiments to measure the power consumption of various key components such as CPU, hard disk and display. Energy consumption characterization in handheld devices for popular GUI platforms from the hardware, software, and application perspectives was discussed in [4]. Such characterization provides a basis for further research on GUI energy optimization. The power consumption of videos encoded using various codec standards and played on handheld devices was presented in [2]. Experiments were performed to explore the effects of the choice of codec standards, file formats, and encoding parameters on power consumed. Since the incorporation of 3D graphics in handhelds poses several serious challenges to the hardware designer, an analysis of the power consumption of mobile 3D graphics pipelines and the effects of various 3D graphics factors such as resolution, frame rate, level of detail, lighting and texture maps on power consumption was presented in [5].

In recent years and improve the energy budget usage of portable units, several energy management techniques have been investigated at different levels of system design – starting from silicon at the bottom to application design at the top, with communication protocols and operating system in between [6 – 9]. The primary objective of these techniques was to find means and methods to reduce the power consumption of mobile devices and extend the battery life time.

Therefore and in view of the slow battery capacity growth, it is becoming increasingly important to develop techniques to achieve high energy efficiency for handheld mobile devices. Designers must strive to maximize the amount of service work that the system can accomplish taking into consideration today's battery capacity constraints. An analysis of the power consumption requirements and an understanding of battery performance as the only energy source in these devices is essential, and offers both users and designers critical information that would assist in efficient system usage and in bringing to the market energy-efficient designs.

Initial and partial results of the work presented here were presented at APCCS [10]. In this paper an expanded and elaborate version of this work is presented. The aim is to experiment with two portable devices, namely the Compaq iPAQ 3950 and the HP iPAQ Rx3715 and assess their battery utilization as well as power consumption characteristics. The methodology used in this work is applicable to most handheld devices.

The rest of the paper is organized as follows: section 2 describes the experiment environment and the experimental setup as well the theoretical framework that forms the basis for the measurements taken. The various loads or benchmark software used is discussed in section

doi:10.4304/jcp.7.1.147-155

3. Experimental results are presented in section 4. The paper is concluded in section 5.

II. EXPERIMENT ENVIRONMENT

The methodology used and the procedural steps taken to collect data that reflects battery usage, and the consumption characteristics are described in the following subsections.

A. Experimental Setting

The experimental setup or testbed consisted of the following components: the pocket PCs, the measurement apparatus and the software or benchmarks to be run to gauge the consumption. The setup is shown in Figure 1.

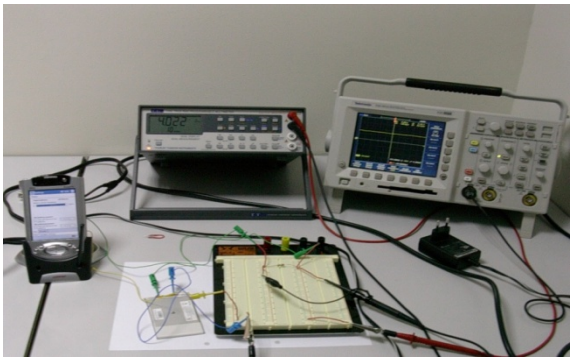


Figure 1. Experimental Setup

The Compaq iPAQ 3950 unit uses an Intel XScale processor. The speed is 400MHz and RAM is 64MB. It comes with an SD/MMC slot for extra memory. It has a transflective TFT display. Audio capabilities include microphone, speaker and a headphone jack. The only mode of wireless communication is through infrared. The operating system used is Windows Pocket PC.

The HP iPAQ Rx3715 unit uses a Samsung 2440 Processor. The speed is 400MHz and 152 MB of memory is available to the user. It also has a SD memory card support for any extra memory if required. The display is a 3.5 inch transflective color QVGA, 240 X 320 pixels, 64K color support. The backlight can be modified so as to change its brightness. For audio capabilities there is an integrated microphone, speaker, 3.5 mm stereo headphone jack, MP3 stereo through audio jack. The communication features include Infrared with a data transfer speed up to 115.2 Kbps, Bluetooth with a 10 meter range and WiFi capabilities. The Operating system used by it is Windows CE.

We used as well a Tektronix TDS 3032 Digital Phosphor Oscilloscope. It has two channels, a bandwidth of 300MHz, and maximum sample rate of 2.5GS/s. It has the capability of acquiring a maximum of 10,000 points waveforms to capture horizontal detail. The signal processing features include averaging the input signal to remove the uncorrelated noise and improve the measurement accuracy. If the signal appears noisy, the waveform intensity knob could be increased to see the noise more easily. The display features include a color LCD and digital phosphor to clearly display intensity modulation in the signal. The oscilloscope also had a built-in floppy disk to store and retreive waveforms and setups. The waveforms could be saved as a picture, spreadsheet file, MathCAD or internal format.

To prevent overcharging of the battery due to a spike upon connecting the Pocket PC directly to the power source, a protection circuit is used. The circuit diagram is of the complete setup is shown in figure 2.

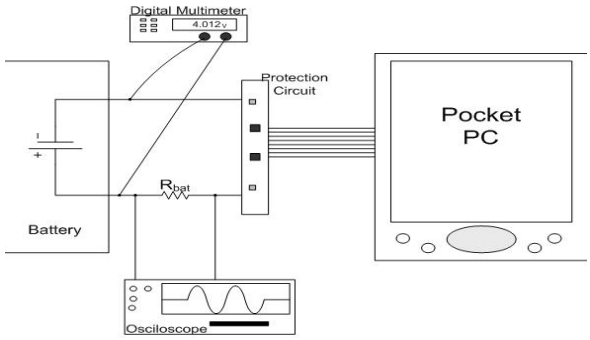


Figure 2. Circuit Diagram

In steady state, the power *P* of the pocket computer can be computed as follows:

$$P = V * I$$

Where V is the battery voltage and I is the current drawn by the computer. In this work, a digital multimeter was used to measure the voltage across the battery terminals. To measure the current, a small resistor R is positioned in series with the lithium battery. The voltage across the resistor V_R is sampled using the oscilloscope; using Kirchoff's law ($I = V_R/R$), we can compute the current drawn by the pocket computer. The instantaneous power is then computed by multiplying this current with the supply voltage. This is similar to the approach utilized in [11] to assess the power characteristics of the Itsy platform.

The oscilloscope was configured to perform a predefined number (*N)* of measurements while calculating the voltage across the battery. In our case, we set *N* at 500. At the end of every acquisition which is a set of *N* measurements the average is calculated.

For a total of N measurements taken Δ units of time apart, an estimate of the average power is computed using the following equation:

$$\mathbf{P_{AVG}} = 1/N \ \Sigma_i^N \ P(i)$$

III. SOFTWARE BENCHMARKS

In this work we experimented with a variety of workloads representing a wide range of tasks and at the same time capable of stressing different components within the PC. The applications used include:

A. The Pocket PC Benchmark v1.02

The application [12] tests the Pocket PC from different points of view such as calculations and heap management

performance. It can execute three types of benchmarks, namely, CPU Benchmark, Graphics Benchmark and Flat File Benchmark. The benchmarks can be executed on a loop count from 1 to 1000. The advantage of this software was that we were able to execute these benchmarks on a loop of 1000 till the battery is drained. A brief description of each benchmark is provided below.

CPU Benchmark -

The CPU benchmark includes the following variations:

Integer arithmetic

The basic four integer arithmetic operations are addition, subtraction, multiplication, and division. Arithmetic operations can be signed or unsigned.

Heap Management

Applications allocate and manipulate memory primarily in their application heap. Space in the application heap is allocated and released on demand. When the blocks in the heap are free to move, the Memory Manager can often reorganize the heap to free space when necessary to fulfill a memory-allocation request. In some cases, however, blocks in the heap cannot move. In these cases, you need to pay close attention to memory allocation and management to avoid fragmenting the heap and running out of memory .

Floating Point Arithmetic

Arithmetic operations on floating point numbers consist of addition, subtraction, multiplication and division.

Graphics Benchmark -

The Graphics Benchmark runs Graphics Device Interface (GDI) applications so as to mimic the Standard graphics functions provided by Windows.

Flat File Benchmark -

The Flat File benchmark helps us study two important aspects of the memory which are file reading and file writing to Pocket PC's internal memory.

B. *The Spb Benchmark 1.0*

The Spb Benchmarks [13] is one of standard benchmarks which are used in Pocket PC power consumption analysis. It offers a variety of tests such as:

- Processor speed
- Memory Bus speed
- Screen and graphics speed
- Battery lifetime
- Storage card read/write speed

Using this software tests can vary from the most economical with no backlight, zero utilization to the most power consuming maximum backlight, video playback. Each single battery test is carried out for the whole battery lifetime period .

Finally, the XCPUScalar software is used to dynamically change the processor speed of the Pocket PC; it unlocks processor dynamic scaling capabilities [14]. We used this application to vary the frequency of the Pocket PCs between 100Mhz to 500MHz to study the effect of the change in frequency on power consumption.

IV. EXPERIMENTAL RESULTS

Using a series of experiments, we run the benchmarks described above using different scenarios and modes of operation to understand effect on battery life and power consumption of the devices. A brief description of each experiment is provided below:

Experiment 1 - We measured the battery consumption when the Pocket PC was idle at normal frequency i.e. 400 MHz and the display brightness was set to maximum.

Experiment 2 - This experiment was performed using one of the battery tests in the SPB benchmark suite. It we played the MP3 with the display off at 400 MHz .

Experiment 3 - In this experiment, MP3 was played at 400 MHz and the display brightness was set to maximum.

Experiment 4 - MP3 is played at the Pocket PCs normal frequency and the display brightness was set to 50%.

Experiment 5 - In this experiment we used the SPB benchmark suite. The battery test that we used periodically opened Pocket Word, loaded a document and closed Pocket Word and the display brightness was set to maximum level. The frequency was set at 400 MHz.

Experiment 6 - In this experiment, we used the CE performance benchmark which executed 1000 loops of CPU intensive operation. The display brightness was set to 0% and the processor frequency was at 400 MHz.

Experiment 7 - In this experiment, MP3 was played at 500 MHz and the display brightness was set to maximum. Also the speaker volume was set to the highest level.

Experiment 8 - In this experiment, MP3 was played at 300 MHz and the display brightness was set to maximum. Also the speaker volume was set to the highest level.

Experiment 9 - In this experiment, MP3 was played at 100 MHz and the display brightness was set to maximum. Also the speaker volume was set to the highest level.

Experiment 10 - In this experiment, we used the 'Pocket PC Benchmark' which executed 1000 loops of CPU intensive operation. The display brightness was set to 0% and the processor frequency was at 500 MHz.

Experiment 11 - In this experiment, we used the CPU benchmark which executed 1000 loops of CPU intensive operation. The display brightness was set to 0% and the processor frequency was at 100 MHz.

Experiment 12 - MP3 was played at 500 MHz and the display brightness was set to 50%.

Experiment 13 - MP3 was played at 100 MHz and the display brightness was set to 50%.

Experiment 14 - In this experiment we used the SPB benchmark suite. The battery test that we used

periodically opened Pocket Word, loaded a document and closed Pocket Word and the display brightness was set to maximum level. The frequency was set at 500 MHz.

Experiment 15 - In this experiment we used the SPB benchmark suite. The battery test that we used periodically opened Pocket Word, loaded a document and closed Pocket Word and the display brightness was set to maximum level. The frequency was set at 100 MHz.

Experiment 16 - In this experiment, we used the 'Pocket PC Benchmark' which executed 1000 loops of Flat File operations. It starts off by performing 1000 file write operations and then 1000 loops of file read operations. We executed this benchmark at 100MHz and the Pocket PC display Brightness was set to 0%.

Experiment 17 - In this experiment, we used the 'Pocket PC Benchmark' which executed 1000 loops of Graphic operation. We set the display brightness to 100%, the frequency was set to 400MHz and executed 1000 loops of graphic operations.

Experiment 18 - In this experiment, we used the 'Pocket PC Benchmark' which executed 1000 loops of Flat File operations. It starts off by performing 1000 file write operations and then 1000 loops of file read operations. We executed this benchmark at 500MHz and the Pocket PC display Brightness was set to 0%.

Experiment 19 - In this experiment, we used the 'Pocket PC Benchmark' which executed 1000 loops of Graphic operation. For the experiment, we set the display brightness to 100%, the frequency was set to 100MHz and executed 1000 loops of graphic operations.

Experiment 20 - In this experiment, we used the 'Pocket PC Benchmark' which executed 1000 loops of Graphic operation. We set the display brightness to 100%, the frequency was set to 500MHz and executed 1000 loops of graphic operations.

Experiment 21 - In this experiment we used the SPB benchmark suite. This benchmark measures the battery lifetime when a video clip is played with media player and the screen backlight was set to maximum. The frequency the Pocket PC was set to was 400MHz.

Experiment 22 - In this experiment, we used the 'Pocket PC Benchmark' which executed 1000 loops of Flat File operations. It starts off by performing 1000 file write operations and then 1000 loops of file read operations. We executed this benchmark at 400MHz and the Pocket PC display Brightness was set to 0%.

Experiment 23 - In this experiment, we kept the Bluetooth ON to study its battery consumption characteristics when its at idle. We used the RX3715 and the frequency was set to the Pocket PC's default frequency which is 400 MHz.

Experiment 24 - In this experiment we used the SPB benchmark suite on the RX3715. This benchmark measures the battery lifetime when a video clip is played with media player and the screen backlight was set to maximum. The frequency the Pocket PC was set to was 400MHz.

Experiment 25 - In this experiment, we left the WiFi adapter ON idle. We used the RX3715 and the frequency was 400 MHz.

Experiment 26 - In this experiment, we used the WiFi adapter to access the internet through the Internet Explorer. We used the RX3715 and the frequency was 400 MHz.

Experiment 27 - In this experiment, we used the Bluetooth to receive files from another Bluetooth device. We used the RX3715 and the frequency was 400 MHz.

Next, we discuss the outcomes and the observations made based on the measurements from these experiments.

Figures 3 through 10 are graphs showing battery discharge versus time when the device is operated under different conditions. For example, for the iPAQ 3950, comparing the results depicted in figures 3 and 4, it is evident that by simply reducing the display brightness to about 50% of the maximum possible, we were able to extend battery life by approximately 50%. The graph of figure 5 clearly shows increasing the frequency of operation in this case comes at a price. When the pocket PC was operated at the lowest frequency (100 Mhz) the battery lasted the longest, however, execution pace was much slower. Operating the device at 500 Mhz drained the battery before even reaching the floating point arithmetic part of the benchmark.

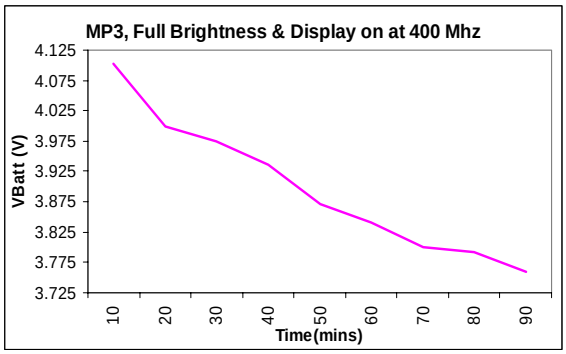


Figure 3. MP3, Full Brightness & Display on at 400MHz

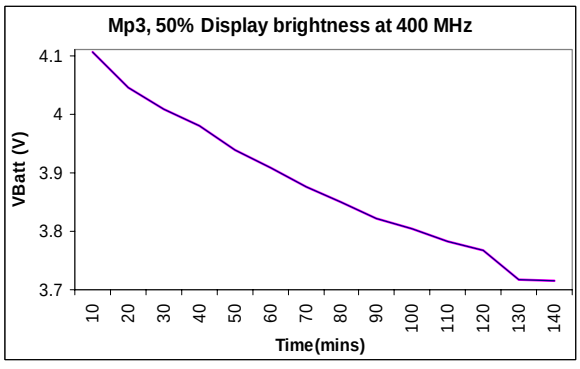


Figure 4. MP3, 50% Brightness at 400 MHz

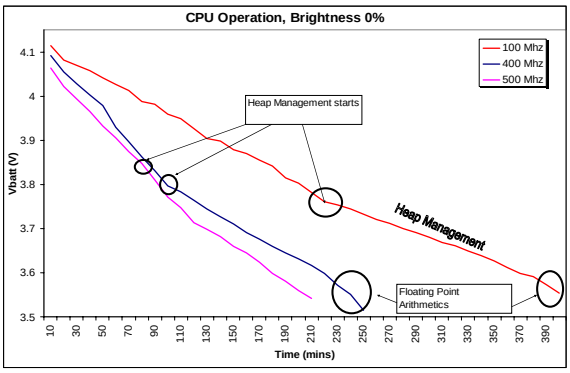


Figure 5. CPU Operation, 0% Brightness at varying frequencies

In figure 6 the SbP benchmark periodically opens "pocket work", loads a document and closes "pocket word" emulating typical pocket PC usage. We kept the brightness at the maximum level and varied the frequency. Operating the device at 400 Mhz yielded optimum results for executing a standard utility, in this case, the 'word application'.

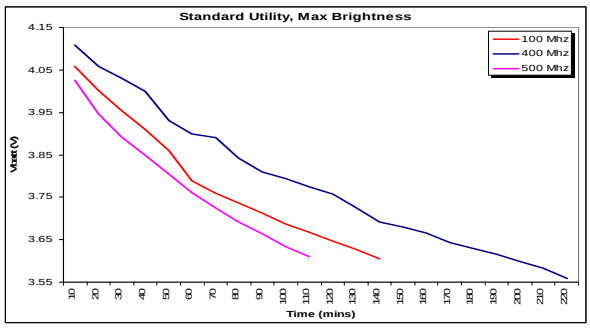


Figure 6. Standard Utility, 100% Brightness at varying frequencies

In figure 7 we used the Flat File bench mark to first write files to the internal memory of the PC, then read the files from internal memory. When writing files to internal memory at 100 MHz and 400MHz battery consumption rate was almost the same, additionally, the same number of jobs is performed in both cases. However, reading files at 400MHz was a better option because it took less power and more time to consume the battery. The beginning of the reading of the files phase has been indicated by circles in Figure 7.

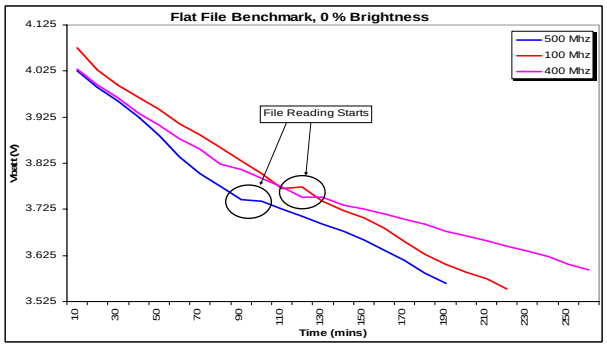


Figure 7. Flat File, 0% Brightness at varying frequencies

Figure 8 depicts the results obtained when we experimented with the WiFi in browsing the net, and using Bluetooth in transferring files from another Bluetooth-enabled device. The graphs are combined for space saving purposes; however, the two protocols are clearly used differently. As expected, Bluetooth has lower power consumption requirements.

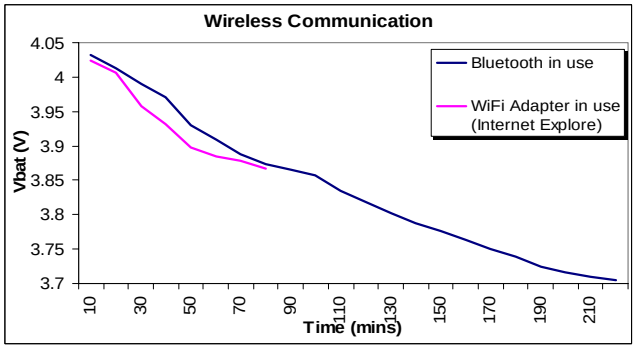


Figure 8. Wireless Communication

Audio performance at 400 Mhz but with different operating scenarios for the display of the device is reflected in figure 9. We can see a very clear difference in how long the battery will last when the display is on and when it is turned off. If the user plays the MP3 with the display off rather than having the display on with only 50% display brightness, he/she would save about 93.568 mW of power. The Pocket PC would run for an additional 4 hours when the display is off.

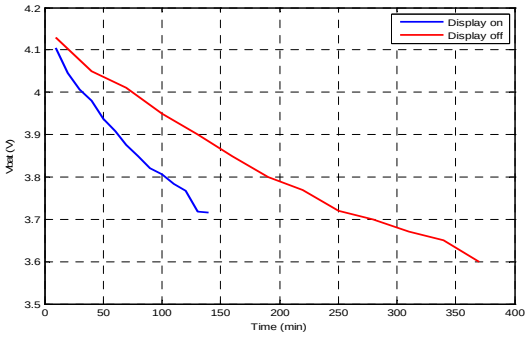


Figure 9. MP3 played at 400MHz

Figure 10 shows the power consumed when the MP3 player is on while the backlight is set to full brightness. The frequency of operation was varied and the power consumed shows that an optimum performance was reached when the device is operated at 400 Mhz.

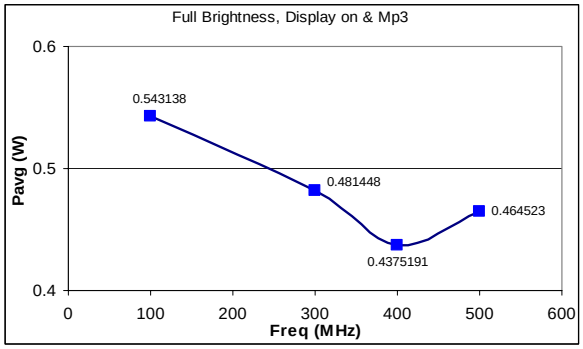


Figure 10. MP3, 100% Brightness at varying frequencies

Tables 1 provides a summary of the various measurements for the iPAQ 3950 unit, and figure 11 is a chart showing its average power consumption when subjected to different operating conditions. Table 2 and figure 12 summarize the results for the HP iPAQ RX3715. Upon examining figure 11, the reader can make interesting observations; for example, operating the device at a slow frequency (in this case 100 Mhz) had an impact only when we consider CPU intensive and graphics operations. On the other hand, for the standard utility test running the device at 400Mhz yielded better results. Operating the device at the 500Mhz did not have a positive impact in most cases.

Benchmark	***Frequency***	***P avg***	***Lifetime***	***Capacity***
	MHz	mW	[Hours]	[mW.h]
Full Brightness & Idle	400	394.218	4.33	1706.966
No display & Mp3	400	177.874	6.33	1125.942
Full Brightness, Display on & Mp3	100	543.138	1.5	814.707
Full Brightness, Display on & Mp3	300	481.448	1.16	558.480
Full Brightness, Display on & Mp3	400	437.519	1.5	656.279
Full Brightness, Display on & Mp3	500	464.523	0.66	306.585
50% Brightness, Display on & Mp3	100	345.041	3	1035.123
50% Brightness, Display on & Mp3	400	271.441	2.33	632.458
50% Brightness, Display on & Mp3	500	349.362	1.83	639.332
Max Brightness & Standard Utility	100	400.955	2.33	934.225
Max Brightness & Standard Utility	400	323.607	3.66	1184.402
Max Brightness & Standard Utility	500	551.299	1.83	1008.877
CPU Operations	100	206.832	7	1447.824
CPU Operations	400	309.436	4.16	1287.254
CPU Operations	500	370.856	3.5	1297.996
Graphics	100	388.793	2.33	905.887
Graphics	400	542.169	1.66	900.001
Graphics	500	547.174	1.33	727.741
0% brightness Flat File Benchmark	100	299.804	5	1499.020
0% brightness Flat File Benchmark	500	328.622	3.16	1038.446

Table 1: Average power and battery life time of the iPAQ 3950. The third column is the total power averaged over a complete battery discharge. The forth column is the average battery lifetime. The fifth is the effective battery capacity (power-lifetime product)

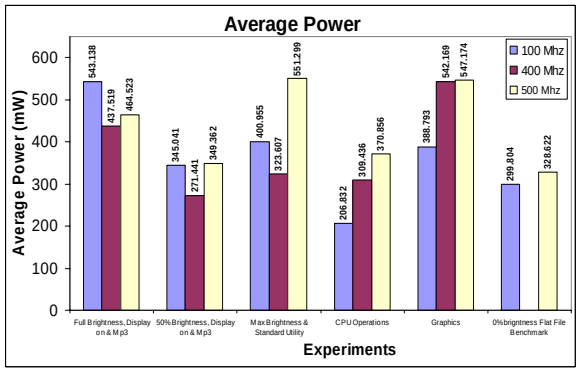


Figure 11. Average Power for iPAQ 3950 using different operating conditions and frequencies

Figure 11 shows the consumption of the HP iPAQ 3950 at 400 Mhz. Two consuming and battery-exhausting scenarios were observed on video play back and WiFi utilization, while the display brightness is set to its maximum. However, the capacity was relatively better in the case of the video playback experiment.

Benchmark	Frequency	P avg	Lifetime	Capacity
	MHz	mW	[Hours]	[mW.h]
Bluetooth idle, Full brightness	400	191.452	8.64	1654.149
Video Playback, Full brightness	400	347.930	1.33	462.747
WiFi Adapter on idle, 0% brightness	400	93.399	10	933.993
WiFi Adapter in use(IE), Max brightness	400	289.197	1.33	384.633
Bluetooth, Max Brightness (receive)	400	232.281	3.66	850.148

Table 2. Average power and battery lifetime of the iPAQ RX3715. The third column is the total power averaged over a complete battery discharge. The forth column is the average battery lifetime. The fifth is the effective battery capacity (power-lifetime product).

V. ANALYSIS

From the results of experiments performed in this work, it is obvious that there are wide variations in battery life time depending on parameters such as frequency, degree of brightness of displays, the requirements of the application running, etc. For example, results summarized in Table 1 Show that operating the MP3 player with a CPU frequency of 400 Mhz, but with the display brightness reduced to 50% there is some savings in battery life (approximately

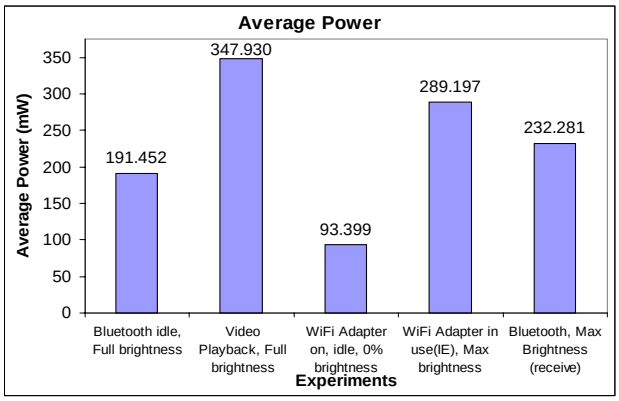


Figure 12. Average Power for iPAQ RX3715 using different operating conditions

15%). At 100 Mhz with the display brightness reduced to approximately 50% from the maximum possible had the effect of doubling the battery life time; power consumption has been reduced by about 36%. Hence, frequency scaling has contributed to a reduction in power consumption and in an extension to battery life. Furthermore, these results indicate that backlight dimming techniques can lead to a substantial increase in battery life, and the use of energy-adaptive display systems enhances the life-time span of the handheld device. These displays can adapt their power consumption based on the contents being displayed on the screen and the area of interest to the user.

Table 1 results also show that energy demands of CPU intensive operations may vary over a wide range. A possible technique that is increasingly being used to reduce CPU power consumption is Dynamic-Voltage and Frequency Scaling or DVFS. Power consumption of CPUs (implemented in CMOS) varies linearly with the frequency and quadratically with the supply voltage and therefore reducing either should lead to a reduction in power consumption. Designers, however, should pay close attention to the inherent tradeoff when using this technique; by reducing the frequency at which the CPU operates, a specific operation will consume less power but may take longer to complete. Although reducing the frequency alone will reduce the average power used by the CPU over that period of time, it may not deliver a reduction in energy consumption overall, because the power savings are linearly dependent on the increased time. While greater energy reductions can be obtained with slower clocks and lower voltages, operations will t ake longer; this exposes a fundamental tradeoff between energy and delay [15]. A graph showing battery-life time changes with frequency for the CPU benchmarks used in this work is provided in figure 13.

It is evident that an increase in frequency has led to a corresponding increase in power consumption and a reduction in battery-life time.

Recent work [16] reported that reducing CPU frequency may not always reduce the energy consumption and that the lowest energy consumption appears at some operating frequency other than the lowest one supported by the processor. In [17], Ikenaga,

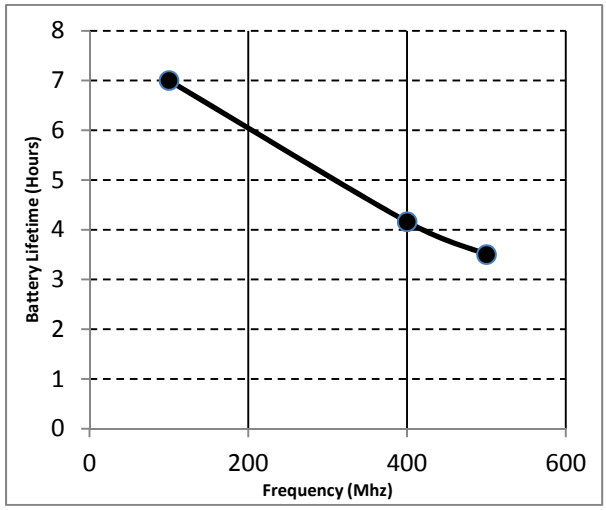


Figure 13. Battery-life Time for CPU operations

defines an MEP (Minimum Energy Point) for mobile devices, such that when operational frequencies are lower than that of the MEP, at which the supply voltage is optimum, energy consumption increases with decreases in operational frequency. In [18], it is argued that to determine an optimal CPU speed-setting, total system power including main memory and battery characteristics must be included and not to consider CPU-power only by itself. A discussion of various DVFS algorithms can be found in [16].

From the graphics benchmark results of Table 1, it is evident there continuous to be a disparity between the energy requirements to produce quality graphics on handheld devices, and the energy available from today's batteries. A major improvement in this area is the introduction of low-power graphics processor (GPU). In [5], a quantitative analysis of workload variations and imbalances of different stages of a mobile 3D graphics pipeline, and the potential for DVFS based power savings is presented. The paper claims that using history-based DVFS strategies can achieve energy savings of up to 50%. A technique to reduce the dynamic power consumption component of a GPU by reducing the precision of its arithmetic operations was discussed in [19]. The recently introduced Tegra 2 mobile processor [20] from NVDIA promises outstanding graphics performance on mobile units. There are no studies available yet in the published literature documenting energy efficiency of this processor.

In this work and using the Flat File benchmark, device memory is subjected to continuous writing and reading operations. In memory banks, power loss can be static or dynamic. Static power loss is technology and processing dependent, while dynamic loss occurs on each instruction or data memory access. That is, energy dissipation is proportional to the amount of data or code being transferred. A reduction in power consumption can be achieved by using code optimization and data compression techniques. Additionally, the use of new memory designs such as CellularRAM and MobileRAM is expected to result in power savings in future handheld devices. CelluarRam is a pseudo SRAM that permits self-

refresh and recharged operations inherent in DRAM technology while providing lower standby and operating currents and lower consumption. In MobileRam an on-chip temperature sensor is used to automatically senses the temperature and choose the most efficient (in terms of power) memory refresh rate [21].

From Table 2, video playback and wi-fi usage are evidently issues of concern to designers when we consider battery utilization. Video decoding has demanding and complex computational requirements and contributes to rapid battery drainage. In [22], it was shown that for handheld devices, increasing resolution cost higher energy needs and often not justified, whereas increasing bit rate gives better picture quality without inducing too much energy consumption.

Since displays are power hungry, an adaptive middleware based approach to optimize backlight power consumption for mobile handheld devices when playing video without compromising quality was discussed in [23].

WiFi radios have a high wakeup and connection maintenance energy, but low energy per bit transmission cost and high bandwidth. Reducing energy consumption is achieved by spending as much time as possible in low power states or power saving modes [24].

Finally, battery-life concerns cannot be addressed without the battery itself as a device being the focus of research at some level. In [1], the authors propose an analytical battery model which can be used for battery life estimation. The model allows designers to predict time-to-failure for a given load and provides a cost metric for life-time optimization algorithms. To our knowledge, this is one of the very few reported work that describes an attempt towards a formal treatment of battery-aware task scheduling and voltage scaling based on a battery model.

VI. CONCLUSION

Portable computing devices continue to come to the market with more powerful processors, enhanced features and increased functionality, and hence, the energy requirements of these devices continue to increase. This increase in energy requirements is not accompanied by a corresponding improvement in battery technology. To develop sound energy management policies, to design energy-efficient units, and to educate the users of these devices in ways and means of prolonging the battery life an examination of their power consumption tendencies is critical.

The aim of this work was to study two battery-powered mobile devices and have an insight into their power consumption requirements and their battery discharge behavior. Through a series of experiments and using different benchmarks we obtained sets of data that highlights the dynamics and the factors that come into play when we discuss battery utilization and energy requirements. Results indicate that factors such as frequency, display brightness, benchmark characteristics, and wireless protocols used have a noticeable impact on battery-capacity and power. It is evident that features such as dynamic-voltage-frequency scaling, and usage of energy-adaptive displays are critical to optimizing battery life-time. A memory subsystem with a hierarchy that reduces the misses in cache and minimizes external reads and writes to external memory is of value as well. Wireless RF designs and protocols that allow for power savings mode will continue to contribute substantially to energy savings. Finally, we argue there is still a need for an energy-management approach to these devices that integrates optimization techniques stemming from cooperation of chip or hardware designers with the software engineers who write the operating systems, compilers, and the applications that run on them.

REFERENCES

[1] D. Rakhmatov and S. Vrudhula, “Energy Management for Battery-Powered Embedded Systems”, ACM Transactions on Embedded Computing Systems, Vol. 2, No. 3, 2003, pp. 277–324.

[2] H. Lin, J.C. Liu and C. W Liao, “Energy Analysis of Multimedia Video Decoding on Mobile Handheld Devices”, Journal of Computer Standards and Interfaces, Vol. 32, Issue 1-2, 2010in

[3] Mahesri and V. Vardhan, “Power Consumption Breakdown on a Modern Laptop”, Intl. Workshop on Power Aware Computing Systems, Dec. 2004

[4] L. Zhong and N. K. Jha, “Graphical User Interface Energy Characterization for Handheld Computers”, Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems

[5] Mochocki, K. Lahiri and S. Cadambi, “Power Analysis of Mobile 3D Graphics”, Proceedings of the conference on Design, automation and test in Europe, 2006

[6] R. Palit, A. Singhand K. Naik, “Modeling the Energy Cost of Applications on Portable Wireless Devices”, Proceedings of the ACM MSWiM Conference, 2008.

[7] S. Mohapatra et al, “Integrated Power Management for Video Streaming to Mobile Handheld Devices”, Proceedings of the eleventh ACM international conference on Multimedia, 2003.

[8] S. Gurun, Priya Nagpurkar, and Ben Y. Zhao, “Energy Consumption and Conservation in Mobile Peer-to-Peer Systems”, Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking, 2006.

[9] Y. Fei et al, “An Energy-Aware Framework for Dynamic Software Management in Mobile Computing Systems”, ACM Transactions on Embedded Computing Systems, Vol. 7, No. 3, Article 27, Publication date: April 2008.

[10] A. Sagahyroon, “Power Consumption in Handheld Computers”, IEEE Asia-Pacific Conference on Circuits and Systems, 2006.

[11] M. Viredez and D. Wallach, “Power evaluation of the Itsy version 2.3”, Technical Note TN57, Digital Western Research Laboratory, Oct. 2000.

[12] http://freewareppc.com/utilities/

[13] http://www.spbsoftwarehouse.com/products/benchmark/.

[14] http://www.immiersoft.com/.

[15] D. Grunwald, P. Levis, and K. Farkas, “Policies for Dynamic Clock Scheduling”, Proceedings of the 4th Conference on Operating Systems Design and Implementation, 2000

[16] Y. Liang, P. Lai, and C. Chiou, “An Energy Conservation DVFS Algorithm for the Android Operating System”, Journal of Convergence, Vol. 1, No. 1, 2010

[17] Y. Ikenaga, "Using Dynamic Voltage Scaling to conserve System Battery Requirements", available at: http://www.eetimes.com/design/embedded/
[18] T. Martin, and D. Siewiorek, "Nonideal Battery and Main Memory Effects on CPU Speed-Setting for Low Power", IEEE Transactions on VLSI, Vol. 9, No. 1, 2001
[19] J. Pool, A. Lastra, and M. Singh, "Energy-Precision Tradeoffs in Mobile Graphics Processing Units", Proceedings of the IEEE International Conference of Computer Design, 2008
[20] "Bringing High-end Graphics to Handheld Devices", white paper available at: www.nvidia.com
[21] N. Sklavos, and K. Touliou, "A System Level Analysis of Power Consumption and Optimization in 3G Mobile Devices" ", Proc. of the 1st Intl. Conf. on New Technologies, Mobility, and Security, 2007
[22] C. H. Lin et al, "Energy Analysis of Multimedia Video Decoding on Mobile Handheld Devices", Journal of Computer Standards and Interfaces, Vol. 32, Issues 1-2, 2010
[23] S. Pasricha et al, "Reducing Backlight Power Consumption for Streaming Video Application on Mobile Handheld Devices", in Proceedings of First Workshop on Embedded Systems for Real Time Multimedia", 2003
[24] G. Anastasi et al, "802.11 Power-saving Mode for Mobile Computing in Wi-Fi Hot Spots: Limitations, Enhancements and Open Issues", Journal Wireless Networks, Vol. 14, No. 6, 2008

Assim Sagahyroon received the Ph.D degree in Electrical Engineering from University of Arizona and a Master Degree also in Electrical Engineering from Northwestern University. He was a tenured faculty with the department of computer science engineering at Northern Arizona University from 1993 to 1999 and then joined the department of math and computer science at California State University. He worked in industry with Lucent and Zhone technologies. He is currently an associate professor with the Department of Computer Science and Engineering at the American university of Sharjah. His research interests include power consumption in VLSI designs and portable devices, hardware description languages, FPGAs, computer architecture and innovative applications of emerging technology.

A Query Verification Scheme for Dynamic Outsourced Databases

Xiaoming Wang
Department of Computer Science, Guangzhou ,China
Email: wxmsq@eyou.com

Duobao Yuan
Department of Computer Science, Guangzhou ,China
Email: dby@eyou.com

***Abstract*—Recently, Kyriakos et al. proposed a Partially Materialized Digest scheme (*PMD*). Their scheme uses separate indexes for the data and the verification information to realize the authenticity and completeness verification in outsourced database. In this paper, we analyze *PMD* and show the drawbacks in *PMD* such as the slow update, the collusion and forgery attacks, and no content access control etc. In order to overcome these defects, the idea of designated verifier signature is introduced into query verification, and a secure, efficient query verification scheme for dynamic outsourced databases is proposed based on *PMD*. The analysis of security and performance shows that the proposed scheme has not only the same advantages with *PMD*, but also prevents the collusion and forgery attacks as well as enforces content access control. Furthermore, the update speed of the proposed scheme is faster than *PMD*, the storage cast of the proposed is lower than *PMD*.**

***Index Terms*—outsourced databases, query authentication, security, efficiency**

I. INTRODUCTION

With rapid developments of Internet applications and network technology, the outsourcing service is emerging as an important new trend called the "application-as-a-service". Outsourcing service is an operation mode that enterprises outsource their information systems or business processes to professional external agencies that can provide these services for much lower cost due to economies of scale. As a recent manifestation of this trend, there have been growing interests in outsourcing database services in both the commercial and the research community. In 2002, Hacigumus et al. first proposed database outsourcing in ICDE (International Conference on Data Engineering)[1]. The basic idea is that data owners delegate their database needs and functionalities to a third-party that provides services to the users of the database. Since a service provider is not always fully trusted, security and privacy of outsourced data are significant issues. These problems are referred to as data confidentiality, query verification, user privacy, and data privacy. Among them, the query verification takes a crucial role to the success of the outsourcing model. The database query verification enables users quickly and accurately to verify if the query results that the database server returned are correct and complete, in order to avoid wrong decisions based on false or incomplete data.

In 2009, Kyriakos et al.[2] proposed a Partially Materialized Digest scheme(*PMD*). Their scheme uses a main index (*MI*) tree solely for storing and querying the data, and a separate digest index (*DI*) tree that contains the *MHT*-based verification information in a compressed form. Because the two processes of the search query result and the verification information can be concurrently executed, therefore their scheme is high efficient. *PMD* can support the completeness and correctness of the query verification, fit to deal with massive data and better support to update the outsourced databases.

This paper analyzes *PMD* and shows the drawbacks in *PMD* such as slow update, the collusion and forgery attacks, and no content access control etc. For dealing with these shortcomings, the idea of designated verifier signature is introduced into query verification, and a secure and efficient query verification scheme for dynamic outsourced databases is proposed based on *PMD*. The analysis of security and performance shows that the improved scheme has not only the same advantages with *PMD*, but also conceals the owner's signatures during the query verification process, prevents collusion and forgery attacks, enforces a certain content access control, and reduces the storage overhead of the database server and improves the update speed etc. Therefore the proposed scheme can overcome the disadvantages of *PMD*.

II. *PMD* ANALYSIS

A. *PMD*[2]

Suppose there are N records in the database, and each record can be expressed in the form of $<key, A_1,\ldots,A_m>$ where *key* is an indexed attribute and $A_i(i=1,2,\ldots,m)$ is a non-indexed attribute. *PMD* uses *MI* tree for storing and querying the data, and *DI* tree for verifying information in a compressed form. *MI* tree is a standard B^+ tree on key, i.e., it does not store any verification information. *DI* tree is essentially an *MHT* with degree 2 and a composite tree. Figure 1[2] illustrates the *DI* tree in a scenario where the database contains N =16 records (r_1 to r_{16}), n = 4

doi:10.4304/jcp.7.1.156-160

external nodes. Where $H(r_i)=H(key|A_1|\ldots|A_m)$, $H_i = H(h_{i|j} \,||\, h_{k|l})$, $h_{i|j} = H(H(r_i)\,|\,H(r_j))$, $S(H_{1|2|3|4})$ is its signed root using ordinary signature scheme such as RSA. $H(.)$ is a hash function.

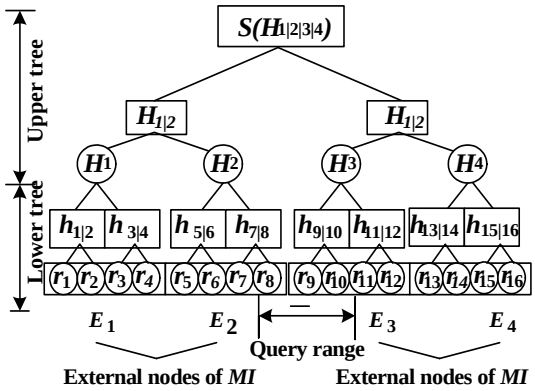


Figure 1 *DI* Digest index example

In order to reduce the storage overhead at the server and the I/O costs for VO (verification object) computation, *PMD* does not materialize the entire *DI* structure, that is, does not store the lower trees. Instead only when an external *MI* node is accessed during query processing and hash values is required from the corresponding lower tree of the *DI* in order to form the *VO*, the server computes them on the fly. Following the same principle, the server does not store the entire upper trees.

In particular, *PMD* divides the n upper tree leaves (i.e., the n lower tree roots) into groups of α. For each group, *PMD* builds a binary *MHT* (suppressed tree). In the second step, *PMD* constructs one binary *MHT* (explicit tree) on top of all the suppressed ones, as shows in Figure 2[2].

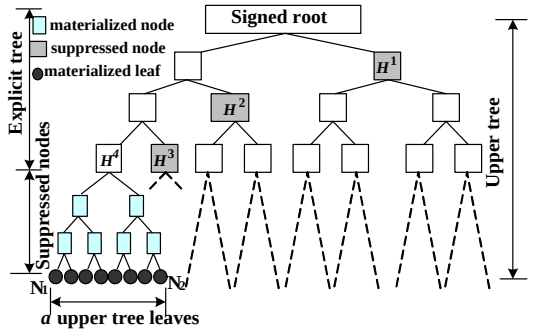


Figure 2 Upper tree compression

For every group (e.g., the group of α leaves shown in the Figure 2), *PMD* forms one *DI* page by inserting following messages (1) the signed *DI* root, and (2) its α hashes (the solid circles), and (3) the sibling digests to its path in the explicit tree (hashes H^1, H^2, H^3, shown as solid squares). In other words, the internal nodes of the suppressed tree (small rectangle) are not materialized, but they are computed on the fly (using the α leaves/solid circles) when needed. Overall, the *DI* contains n/α disk pages. Thus, *PMD* can compute *VO* for any range query with at most 2 disk accesses, one for each boundary object. If both c *left* and *right* are covered by the same suppressed tree, then only one page access is needed for their common *DI* block.

The *DI* works like an ordinary *MHT*. *PMD* considers the paths that lead to the left and to the right boundary objects as *left* and *right*. The *VO* for a range query contains (1) the signed *DI* root, (2) all left sibling hashes to the path of *left*, and (3) all right sibling hashes to the path of *right*. Upon receipt of the query result, the user combines it with *VO* components (2) and (3), to reconstruct the (missing) part of the *DI* between the paths of *left* and *right*. Then, the user verifies with the owner's public key whether the root of *DI* (i.e., component (1) of the *VO*) matches the locally computed root hash. If they match, the query results are deemed both complete and authentic; the collision-resistance of the hash function ensures that it is computationally infeasible for the server to tamper with the result and yet manage to produce hashes that match the original ones.

B. PMD analysis

Comparing with previous query verification scheme *PMD* has the following advantages: (1) there is more flexible architecture. When the user does not need verification, then *PMD* is similar to an ordinary and non-authenticated index without any performance degradation due to verification structure. Moreover, existing database can be directly authenticated, without having to be modified and re-uploaded. Therefore, the construction cost is significant savings. Also, query answering and *VO* generation can be delegated to different servers, thus the server's performance is full used; (2) separating the query processing and the *VO* computation tasks, this provides that the search result and the verification process can be implemented in parallel, so response is faster.

However, we analyze *PMD* and discover that *PMD* has the following shortcomings:

1) There exists the forgery attack in *PMD*. *PMD* uses an ordinary signature scheme such as RSA to sign *DI* root. When *PMD* deals with a user query, server returns the query results and the authentication objects including the data owner's signature to the user, thus exposes the data owner's signature. The ordinary signature is self-proven and can be passed. Anyone can verify the correctness of the signature and data as long as access to the signature. If an attacker intercepts the data and verification information that are uploaded to the server by data owner, he can impersonate the server to provide services for users or distribute the data owner's data and signatures. If a user already obtains the data and the authentication objects including the data owner's signature, he can impersonate the server to provide services for users or distribute the data owner's data and signatures. It is not permitted in practical applications such as payment services through the server.

2) There exists a collusion attack in *PMD* and it cannot achieve the content access control. When *PMD* deals with a user query, the server directly gives query results and the authentication objects including the data owner's

signature to a user, thus the users can make collusion attack together. For example, there are two legitimate users *A* and *B*, the data owner grants different privileges to *A* and *B* such as *A* can only query the index values in the range of $\alpha \le key \le b$ and *B* can only query the index values in the range of $b \le key \le c$. After *A* and *B* obtain their data and the authentication objects through legitimate query, they can collude and impersonate the server to provide services for other users such as accessing the data in the range of $\alpha \le key \le c$. However, *A* or *B* has no right to query the index values in the range of $\alpha \le key \le c$.

3) *PMD* can not well protect the privacy of data. When the user needs or only views some of these fields, the server must return all of these fields to the user in order to enable the user to verify the correctness and completeness of the data. Thus *PMD* leaks some unnecessary fields to the user and does not protect the privacy of data and implement the column-based access control, which is not allowed in many applications.

4) The update speed of *PMD* is slow. As long as the data is updated, the root node must be re-signed. Because each *DI* page stores a signed *DI* root, so each update (object insertion or deletion) requires modification of all *DI* pages, thus the cost of update is high. Although *Kyriako* et al.[2] proposed the dynamic *PMD*(*dPMD*) to resolve this problem, but *dPMD* still needs (2*d*-1) disk accesses (where *d* is the high of *DI* tree). When the database stores huge amounts of data such as $2d-1 >> 2$, *dPMD* needs to access the disk many times.

III. A NEW QUERY VERIFICATION SCHEME

In order to overcome the disadvantages in *PMD*, we improve *PMD* and propose a new query verification scheme for dynamic outsourced databases. We modify *PMD* as following:

1) Similar to *PMD*, our scheme uses *MI* tree for storing and querying the data, and *DI* tree for verifying information in a compressed form. But in order to improve the update speed, our scheme does not save the signed *DI* root in each *DI* page and stores the signed *DI* root in a separate disk block, which is different from *PMD*. When the owner inserts/deletes an object in the database, *PMD* requires to modify all *DI* pages and needs many I/O operations, however our scheme does not require to modify all *DI* pages, and needs the times of I/O operations much less than *PMD*. Therefore, the update speed of our scheme is fast than *PMD*.

Dealing with user query for the first time, server may read the signed *DI* root from the disk and save the signed *DI* root in memory until the signed *DI* root needs to be updated, thus our method guarantees the fact that server needs 3 disk accesses for only dealing with user query for the first time, and needs 2 disk accesses for each subsequent user queries, which just is the same as *PMD*. Therefore, the response speed of our scheme is as fast as *PMD*

Consider the examples in Figure 3 and Figure 4, let $|S|$ is the size of the signed *DI* root, $|H|$ is the size of hash value. Assuming that a disk block fits 7 hashes (i.e. $B=7|H|$), and $|S|=3|H|$.

In *PMD*, a *DI* tree showed in Figure 3 can be divided into 8 suppressed trees (H^1- H^8), i.e. $\alpha = 4$, because a *DI* page contains 4 hashes (the solid circles), and the sibling digests to its path in the explicit tree such as hashes H^2, H^9, H^{10} (shown as solid squares), in other words, the server needs 8 disk blocks to store the *DI* tree.

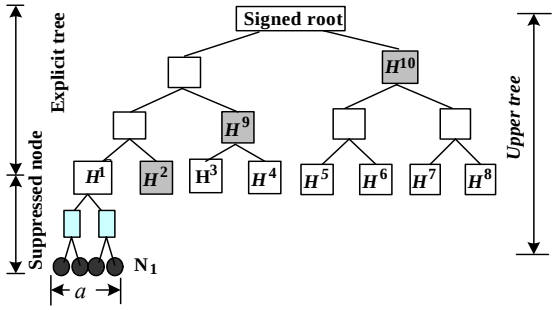


Figure 3 *DI* tree in *PMD*

If the owner inserts/deletes an object corresponding to the external leaf N_1, the updated value of this leaf causes the digest changes that propagate upwards until the root of the *DI* tree. Because each *DI* page materializes the signed *DI* root, therefore, server needs to modify all *DI* blocks and 8 I/O operations to reflect the update.

However, in our scheme, a *DI* page does not materialize the signed *DI* root, and only contains corresponding hashes, the signed *DI* root alone is stored in another a disk block. A *DI* tree showed in Figure 4 can be divided into 4 suppressed trees (H^1- H^4), i.e. $\alpha = 8$, server needs 4 disk blocks to store the *DI* tree.

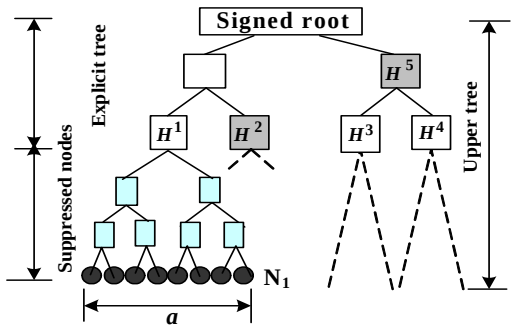


Figure 4 *DI* tree in our scheme

If the owner inserts/deletes an object corresponding to the leaf N_1, the updated value of this leaf causes digest changes that propagate upwards until the root of the *DI*. Because each *DI* page does not materialize the signed *DI* root, and only 3 *DI* pages relate with H^1hashes, therefore, server needs to modify 3 *DI* blocks and 3 I/O operations to reflect the update. Therefore, the update speed of our scheme is fast than *PMD*.

2) In *PMD* and some query verification technology [2~6,8~10], using ordinary signature such as RSA signs the root of *DI* tree, thus brought security issues such as forgery and collusion attacks as well as no content access

control etc. For dealing with these problems, we construct a new signature scheme combining with the idea of designated verifier, and use the new signature scheme to sign the root of *DI* tree, and build new query verification scheme of outsourced databases. When server deals with user queries, the server does not disclose the data owner's signature. Therefore, the new query verification scheme can prevent collusion and forgery attacks, and implement content access control.

New verification scheme is as following:

Let p_1, p_2, p_1' , p_2' and q be distinct large primes, and $p_1 = 2qp_1' + 1$, $p_2 = 2qp_2' + 1$,$n=p_1p_2$, $\varphi(n)$ =(p_1-1)(p_2-1) , g be the primitive element in $GF(n)$. $h(.)$ is a secure one-way hash function.

The data owner completes the following steps to sign root h_{root}.

1) The data owner computes two pair of secret keys (e_o, d_o) and (e_r,d_r) using RSA algorithm, and sends (e_r,d_r,g,n) to server.

2) The data owner chooses a random number $\sigma \in Z_n^*$, computes

$$Y_r = g^{e_r} \bmod n \text{ , } x = Y_r^{\sigma} \bmod n \text{ ,}$$
$$s = (\sigma d_o - h(h_{root})e_o) \bmod q$$

and sends (s, x) to server.

User verifies the query results:

1) Upon receiving the user access request, server chooses a random integer $\beta \in Z_n^*$ and computes $\gamma = \beta^{e_r} \bmod n$, and sends γ to the user.

2) The user chooses a random number $v \in Z_n^*$, and sends v to server.

3) On receiving v, server computes $Y_o = g^{e_o} \bmod n$ and

$$x^{d_r} = [(g^{e_r})^{\sigma}]^{d_r} = g^{\sigma} \bmod n \text{ , } z = \beta(g^{\sigma})^{v} \bmod n$$

and sends (z, s) and the query results to the user.

4) The user verifies the results of query. According to the principle of *MHT*, the user can calculate the hash value of the root based on the query results and *VO*, thus the user can verify

$$z^{e_r} = \gamma(g^s Y_o^{h(h_{root}^{'})})^{e_o v e_r} \bmod n$$

If it holds, then the query results are authentic and complete. Because

$$z^{e_r} = \beta^{e_r}(g^{\sigma})^{ve_r} = \gamma(g^s g^{h(h_{root})e_o})^{e_o v e_r}$$
$$= \gamma(g^s Y_o^{h(h_{root})})^{e_o v e_r} \bmod n$$

In the above query verification process, the user must cooperate with server to complete the verification of query results. Therefore, even if a attacker steals data files from the server or intercepts all the information on the process of the data owner uploading the data and verify information to server, the attacker can not impersonate the server to provide services for users since the attacker does not know the server's private key d_r and can not calculate

$$x^{d_r} = [(g^{e_r})^{\sigma}]^{d_r} = g^{\sigma} \bmod n$$

Therefore, our query verification scheme can resist the forgery attack existed in *PMD*. Moreover, because the server returned the different signature to the user each time, the user cannot use an existing signature to implement the collusion attack. So our scheme can also resist the collusion attack and achieve the content access control.

3) Aiming at the defect that *PMD* cannot realize the column-based access control, our scheme uses

$$H(r_i)= H(key||H(A_1)||\ldots||H(A_m))$$

to compute the message authentication code of the record r_i. When a user queries some fields, the server only returns the values of query fields and the hash values of other fields to the user, and does not return to the values of all fields, and hence does not leak unnecessary information to the user. Thus our scheme protects the privacy of data and implements the column-based access control. Therefore, our scheme can overcome the defect that *PMD* cannot implement the column-based access control.

PERFORMANCE AND SECURITY ANALYSIS

Our scheme, which is similar to *PMD*, uses *MI* tree for storing and querying the data, and *DI* tree for verifying information in a compressed form. But our scheme is different from *PMD*. Our scheme does not save the signed *DI* root in each *DI* page and stores the signed *DI* root in a separate disk block, thus improves update speed. Moreover, our scheme reduces the server's storage overhead since a signed *DI* root is stored.

After start, server immediately reads the signed *DI* root from the disk and computes (Y_o , g^{σ}), and then stored them in memory. In order to compute a *VO*, server needs 2 disk accesses for each user query, which just likes *PMD*. Therefore, the response speed of our scheme is as fast as *PMD*.

In order to verify the storage cost and update speed of our scheme, we perform experiments using Forest CoverType Data set that is provided by University of California, Irvine. Our experiments focus on compare between our scheme and *PMD*. The server storage cost simulation shows as Figure 5. The update speed simulation shows as Figure 6.

The experiments show that our scheme reduces the storage overhead of the database server and improves the update speed at the database server.

We introduce the idea of designated verifier to the query verification technology of outsourced databases, and construct a new signature scheme. We use the new signature scheme to sign the root of *DI* tree, rather than the ordinary signature scheme. Because the server returns to user different signatures when handling users query each time, the data owner's signature is hidden. Thus the attackers or users cannot impersonate the server to provide services for other users by using the existing signatures, and some users cannot make collusion attack using their own data and verification information. Therefore, our scheme overcomes the disadvantages of *PMD* such as collusion attack and forgery attack. The previous query verification schemes also use the ordinary signature scheme such as RSA signature scheme, but the

ordinary signature is self-proven and can be passed. Anyone can verify the signature as long as access to the signature. Therefore these schemes also exist above problems. We introduce the idea of designated verifier to the query verification technology of outsourced databases, and well resolve these security issues.

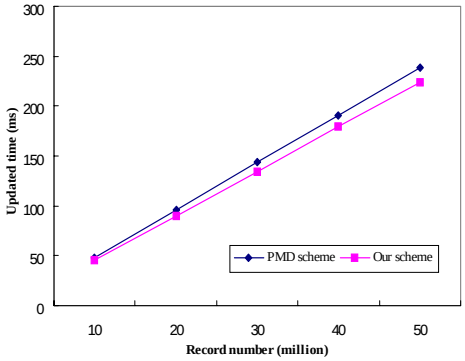


Figure 5 Server storage cost

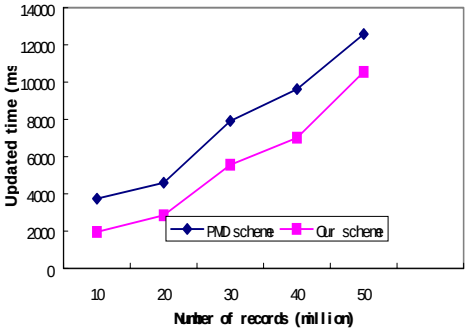


Figure 6 Server update speed

Because using $H(r_i)=H(key||H(A_1)||\ldots||H(A_m))$ to compute the message authentication code of the record r_i, server only returns the values of query fields and the hash values of other fields to the user when a user queries some fields. Therefore, our scheme can better support the projection queries, implement the content access control, and protect private data by the greatest extent.

V. CONCLUSION

In this paper, we improve *PMD*, and propose a new query verification scheme. We modify the storage contents of the *DI* page in *PMD* in order to improve the update speed. We also construct a new signature scheme combining with the idea of designated verifier signature, and replace the signature scheme of *PMD* with the new signature scheme to sign *DI* root in order to resist collusion and forgery attacks. We use $H(r_i)=H(key||H(A_1)||\ldots||H(A_m))$ to compute the message authentication code of the record r_i in order to protect the privacy of data and implement the column-based access control. We perform experiments on our scheme, and focus on comparing our scheme with *PMD*. The result of the experiments shows that the update speed of our scheme is faster than *PMD*, the storage cast of our scheme is lower than *PMD*.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China under Grant (61070164); Science and Technology Planning Project of Guangdong Province, China (S2011010001599); Natural Science Foundation of Guangdong Province, China (81510632010 0000 22).

REFERENCES

[1] H.,Hacigumus,S.Mehrotra,B.Iyer, et al. Providing database as a service. ICDE'02 Proceedings of the 18th International Conference on Data Engineering, IEEE Computer Society Washington, pp. 29-38.

[2] M.Kyriakos,S.Dimitris,P.HweeHwa. Partially Materialized Digest Scheme: An Efficient Verification Method for Outsourced Databases. The VLDB Journal,2009,18(1),pp. 363–381.

[3] P.Devanbu, M.Gertz, C.Martel, et al. Authentic third-party data publication. 14th IFIP 11.3 Working Conference in Database Security, The Netherland: Kluwer, 2000,pp.101-112.

[4] M.Charles, N.Glen, D.Premkumar, et al. A general model for authenticated data structures. Algorithmica.2004,39, pp.21-41.

[5] E.Mykletun, M.Narasimha, G.Tsudik. DSAC: integrity for outsourced databases with signature aggregation and chaining. ACM CIKM. New York, 2005,pp.235-236.

[6] L.Feifei,H.Marios, K.George,et al. Dynamic Authenticated Index Structures for Outsourced Database. Chicago, Illinois: ACM SIGMOD, 2006,pp.121-132.

[7] E. Mykletun, M.Narasimha, G.Tsudik. Authentication and Integrity in Outsourced Databases. ACM Transactions on Storage,2006,2(2), pp. 107-138.

[8] N.Maithili,T.Gene. Authenticated of Outsourced Databases Using Signature Aggregation and Chaining. Wuwongse: DASFAA. 2006, LNCS 3882,pp.:420-436.

[9] J.A.Mikhail,C.Y.Sun,K.Ashish. Efficient Data Authentication in an Environment of Untrusted Third-Party Distributors. ICDE 2008,pp.696-704.

[10] T.G.Michael,T.Roberto,T.Nikos.Superefficient verification of dynamic outsourced databases. CT-RSA 2008, LNCS 4964,pp.407-424.

[11] L.Bouganim, F.D.Ngoc, P.Pucheral,et al. Chip-secured data access: Reconciling access rights with data encryption. Proc. of the 29th International Conference on Very Large Database. VLDB Endowment, 2003,pp.1133-1136.

[12] L. Bouganim, C.Cremarenco, F.D.Ngoc, et al. Safe data sharing and data dissemination on smart device. Proc. of the 2005 ACM SIGMOD International Conference on Management of Data. New York: ACM Press. 2005, pp.888-890.

[13] Q Zhu, J.H. Chen, J.J.Le. Digital watermark-based authentication of queries in outsourced database. Journal of Computer Applications. 2008, 28(3),pp.605-608.

Xiaoming Wang received her Ph.D Degree from the College of Mathematics, Nankai University, China, in 2003. Currently she is a professor in the Computer Science Department, Jinan University. Her research interests include database security, network security, cryptography, etc.

Duobao Yuan received his M.S. Degree from Department of Computer Science, Jinan University, China, in 2010. Currently he is a engineer in Commercial Bank of China. His research interests include database security, network security,etc.

Neighborhood Component Feature Selection for High-Dimensional Data

Wei Yang,Kuanquan Wang and Wangmeng Zuo
Biocomputing Research Centre, School of Computer Science and Technology,
Harbin Institute of Technology, Harbin, 150001
Email: wangkq@hit.edu.cn

***Abstract*— Feature selection is of considerable importance in data mining and machine learning, especially for high dimensional data. In this paper, we propose a novel nearest neighbor-based feature weighting algorithm, which learns a feature weighting vector by maximizing the expected leave-one-out classification accuracy with a regularization term. The algorithm makes no parametric assumptions about the distribution of the data and scales naturally to multiclass problems. Experiments conducted on artificial and real data sets demonstrate that the proposed algorithm is largely insensitive to the increase in the number of irrelevant features and performs better than the state-of-the-art methods in most cases.**

***Index Terms*— feature selection, feature weighting, nearest neighbor**

I. INTRODUCTION

With the emergence of a great quantity of high dimensional data in various applications, including information retrieval, automated text categorization, combinatorial chemistry and bioinformatics, feature selection has become more and more important in data mining and machine learning [1]. Feature selection is the technique of selecting a small subset from a given set of features by eliminating irrelevant and redundant features. Proper feature selection not only reduces the dimensions of features and hence amount of data used in learning, but also alleviates the effect of the curse of dimensionality to improve algorithms' generalization performance. Furthermore, it also increases the execution speed and the models' interpretability.

Generally speaking, feature selection algorithms now usually fall into one of the three categories [2]: filter, wrapper and embedded methods. In the filter model, feature selection is done by evaluating feature subset with the criterion functions characterizing the intrinsic properties of the training data, such as interclass distance (e.g., Fisher score), statistical measures (e.g., Chi-squared) and information theoretic measures, not involving the optimization of performance of any specific classifier directly. On the contrary, the last two methods are closely related with specified classification algorithms and perform better than filter methods in most cases. The wrapper model requires one predetermined classifier in feature selection and uses its performance to evaluate the goodness of selected feature subsets. Since the classifier need always be trained for each feature subsets considered, wrapper methods are computationally intensive and thus often intractable for large-scale feature selection problems. In the embedded model, feature selection is built into the classifier construction and gradient descent method is usually used to optimize the feature weights, which indicate the relevance between the corresponding features and the target concept. The advantages of the embedded methods are that they are not only less prone to overfitting but also computationally much more efficient than wrapper methods. In particular, many SVM-based embedded methods have been proposed [3], [4]. More comprehensive reviews on feature selection methodologies can be referred to [2], [5], [6].

Nearest neighbor is a simple and efficient nonlinear decision rule and often yields competitive results compared with the state-of-the-art classification methods, such as support vector machines and neural network. Recently, several nearest neighbor-based feature weighting methods, including RELIEF [7], Simba [8], RGS [9], I-RELIEF [10], LMFW [11], Lmba [12] and FSSun [13], have been successfully developed and shown the better performance on high-dimensional data analysis. Inspired by the previous work [14], we propose a novel nearest neighbor-based feature selection method called neighborhood component feature selection (NCFS) algorithm. The proposed algorithm uses gradient ascent technique to maximize the expected leave-one-out classification accuracy with a regularization term. Experiments conducted on artificial and real data sets show that NCFS is almost insensitive to the increase in the number of irrelevant features and performs better than Simba, LMFW and FSSun in most cases.

The rest of the paper is organized as follows. Section 2 proposes a novel feature selection algorithm based on neighborhood component. Section 3 summarizes some related feature selection approaches. Experiments conducted on toy data and real microarray datasets to evaluate the effectiveness of the proposed algorithm are presented in Section 4. Finally, conclusions are given in Section 5.

II. NEIGHBORHOOD COMPONENT FEATURE SELECTION

Let $T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_N, y_N)\}$ be a set of training samples, where $\mathbf{x}_i$ is a d-dimensional feature

This work is supported in part by the National Natural Science Foundation of China under No.s 60872099 and 60902099.

doi:10.4304/jcp.7.1.161-168

vector, $y_i \in \{1, \ldots, C\}$ is its corresponding class label and N is a number of samples. The goal is to find a weighting vector w that lends itself to select the feature subset optimizing nearest neighbor classification. In terms of the weighting vector $\mathbf{w}$, we denote the weighted distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ by:

$$D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{d} w_l^2 |x_{il} - x_{jl}| \tag{1}$$

where w_l is a weight associated with lth feature.

For nearest neighbor classification to succeed, an intuitive and effective strategy is to maximize its leave-one-out classification accuracy on the training set T. However, due to the true leave-one-out accuracy that selects nearest neighbor as a classification reference point is a non-differentiable function, an effective approximation is that the reference point is determined by a probability distribution. Here, the probability of $\mathbf{x}_i$ selects $\mathbf{x}_j$ as its reference point is defined as:

$$p_{ij} = \begin{cases} \frac{\kappa(D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j))}{\sum_{k \neq i} \kappa(D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_k))}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases} \tag{2}$$

where $\kappa(z) = \exp(-z/\sigma)$ is a kernel function and the kernel width σ is an input parameter that influences the probability of each points being selected as the reference point. In particular, if $\sigma \to 0$, only the nearest neighbor of the query sample can be selected as its reference point. On the other hand, if $\sigma \to +\infty$, all the points have the same chance being selected apart from the query point. Based on above definition, the probability of the query point $\mathbf{x}_i$ being correctly classified is given by:

$$p_i = \sum_j y_{ij} p_{ij} \tag{3}$$

where $y_{ij} = 1$ if and only if $y_i = y_j$ and $y_{ij} = 0$ otherwise. Therefore, the approximate leave-one-out classification accuracy can be written as:

$$\xi(\mathbf{w}) = \frac{1}{N} \sum_i p_i = \frac{1}{N} \sum_i \sum_j y_{ij} p_{ij} \tag{4}$$

Note that when $\sigma \to 0$, $\xi(\mathbf{w})$ is the true leave-one-out classification accuracy. Moreover, in order to perform feature selection and alleviate overfitting, we further introduce a regularization term and hence obtain the following object function:

$$\xi(\mathbf{w}) = \sum_i \sum_j y_{ij} p_{ij} - \lambda \sum_{l=1}^{d} w_l^2 \tag{5}$$

where $\lambda > 0$ is a regularization parameter which can be tuned via cross validation. Note that the coefficient $1/N$ in (4) is neglected since it only means that the parameter λ makes a corresponding change and the final solution vector is the same. Since the object function $\xi(\mathbf{w})$ is differentiable, its derivative with respect to w_l can be computed:

$$\begin{aligned} \frac{\partial \xi(\mathbf{w})}{\partial w_l} &= \sum_i \sum_j y_{ij} \left[\frac{2}{\sigma} p_{ij} \left(\sum_{k \neq i} p_{ik} |x_{il} - x_{kl}| - |x_{il} - x_{jl}| \right) w_l \right] - 2\lambda w_l \\ &= \frac{2}{\sigma} \sum_i \left(p_i \sum_{k \neq i} p_{ik} |x_{il} - x_{kl}| - \sum_j y_{ij} p_{ij} |x_{il} - x_{jl}| \right) w_l - 2\lambda w_l \\ &= 2 \left(\frac{1}{\sigma} \sum_i \left(p_i \sum_{j \neq i} p_{ij} |x_{il} - x_{jl}| - \sum_j y_{ij} p_{ij} |x_{il} - x_{jl}| \right) - \lambda \right) w_l \end{aligned} \tag{6}$$

Using the above derivative leads to the corresponding gradient ascent update equation. We name the proposed algorithm as neighborhood component feature selection (NCFS) and its pseudocode is given in Algorithm 1. It should be noted that we didn't use the line search method to identify the step length α in the iteration. The main reason is that the evaluation of the objective function requires expensive computation. In Algorithm 1, the update of α is determined by our experience. However, we found that it works well in practice.

Algorithm 1 Neighborhood Component Feature Selection

procedure NCFS($T, \alpha, \sigma, \lambda, \eta$) ▷ T: training set, α: initial step length, σ: kernel width, λ: regularization parameter, η: small positive constant;
  Initialization: $\mathbf{w}^{(0)} = (1, 1, \ldots, 1), \epsilon^{(0)} = -\infty, t = 0$
  repeat
    for $i = 1, \cdots, N$ **do**
      Compute p_{ij} and p_i using $\mathbf{w}^{(t)}$ according to (2) and (3)
    for $l = 1, \cdots, d$ **do**

$$\Delta_l = 2 \left(\frac{1}{\sigma} \sum_i \left(p_i \sum_{j \neq i} p_{ij} |x_{il} - x_{jl}| - \sum_j y_{ij} p_{ij} |x_{il} - x_{jl}| \right) - \lambda \right) w_l^{(t)}$$

    $t = t + 1$
    $\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \alpha \boldsymbol{\Delta}$
    $\epsilon^{(t)} = \xi(\mathbf{w}^{(t-1)})$
    if $\epsilon^{(t)} > \epsilon^{(t-1)}$ **then**
      $\alpha = 1.01\alpha$
    else
      $\alpha = 0.4\alpha$
  until $|\epsilon^{(t)} - \epsilon^{(t-1)}| < \eta$
  $\mathbf{w} = \mathbf{w}^{(t)}$
  return w

III. Related Work

A. Simba

Simba [8] is a feature weighting algorithm based on the hypothesis-margin of 1-NN. For each sample $\mathbf{x}_i$ in the training set T, its hypothesis-margin is given by:

$$\theta_{\mathbf{w}}(\mathbf{x}_i) = \frac{1}{2} \left(\|\mathbf{x}_i - \mathrm{NM}(\mathbf{x}_i)\|_2 - \|\mathbf{x}_i - \mathrm{NH}(\mathbf{x}_i)\|_2 \right) \tag{7}$$

where $\|z\|_2 = \sqrt{\sum_l w_l^2 z_i^2}$, $\mathrm{NM}(\mathbf{x}_i)$ and $\mathrm{NH}(\mathbf{x}_i)$ are the nearest neighbor of $\mathbf{x}_i$ from the different class and the same class under the weighting feature space, respectively. In order to maximize the margins of all the samples, the objective function of Simba is defined as:

$$\varepsilon(\mathbf{w}) = \sum_{i=1}^{N} u(\theta_{\mathbf{w}}(\mathbf{x}_i)) \tag{8}$$

where $u(z)$ is the utility function, which controls the contribution of each margin term to the total sum. In general, the linear utility function $u(z) = z$ and the sigmoid utility $u(z) = 1/(1+\exp(-\beta z))$ are often used. Note that the regularization is not considered in this model. Therefore,Simba is often prone to overfitting when the training set size is small relative to the dimension of the input.

B. FSSun

FSSun [13] is also a feature weighting algorithm based on the hypothesis-margin.In the algorithm,the Manhattan distance instead of the Euclidean distance is used to define the margin and the nearest neighbors of a given sample are considered as hidden variables. The expectation hypothesis-margin of sample $\mathbf{x}_i$ based on the hidden variables can be written as:

$$\begin{aligned}\rho_{\mathbf{w}}(\mathbf{x}_i) &= \mathrm{E}_{k\sim M_i}\left(\|\mathbf{x}_i-\mathbf{x}_k\|_{\mathbf{w}}\right) - \mathrm{E}_{k\sim H_i}\left(\|\mathbf{x}_i-\mathbf{x}_k\|_{\mathbf{w}}\right)\\ &= \textstyle\sum_{k\in M_i} P\left(\mathbf{x}_k=\mathrm{NM}(\mathbf{x}_i)|\mathbf{w}\right)\mathbf{w}^T|\mathbf{x}_i-\mathbf{x}_k|\\ &\quad - \textstyle\sum_{k\in H_i} P\left(\mathbf{x}_k=\mathrm{NH}(\mathbf{x}_i)|\mathbf{w}\right)\mathbf{w}^T|\mathbf{x}_i-\mathbf{x}_k|\\ &= \mathbf{w}^T\big(\textstyle\sum_{k\in M_i} P\left(\mathbf{x}_k=\mathrm{NM}(\mathbf{x}_i)|\mathbf{w}\right)|\mathbf{x}_i-\mathbf{x}_k|\\ &\quad - \textstyle\sum_{k\in H_i} P\left(\mathbf{x}_k=\mathrm{NH}(\mathbf{x}_i)|\mathbf{w}\right)|\mathbf{x}_i-\mathbf{x}_k|\big)\\ &= \mathbf{w}^T\mathbf{z}_i\end{aligned} \tag{9}$$

where $M_i = \{k : 1\le k\le N, y_k \ne y_i\}$, $H_i = \{k : 1\le k\le N, y_k = y_i\}$, $|\cdot|$ is an element wise absolute operator, $P\left(\mathbf{x}_k=\mathrm{NM}(\mathbf{x}_i)|\mathbf{w}\right)$ and $P\left(\mathbf{x}_k=\mathrm{NH}(\mathbf{x}_i)|\mathbf{w}\right)$ are the probabilities of sample $\mathbf{x}_k$ being the nearest miss or hit of $\mathbf{x}_i$, respectively. These probabilities are defined as:

$$P\left(\mathbf{x}_k=\mathrm{NM}(\mathbf{x}_i)|\mathbf{w}\right) = \frac{\kappa(\|\mathbf{x}_i-\mathbf{x}_k\|_{\mathbf{w}})}{\sum_{j\in M_i}\kappa(\|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{w}})}, \forall k\in M_i \tag{10}$$

And

$$P\left(\mathbf{x}_k=\mathrm{NH}(\mathbf{x}_i)|\mathbf{w}\right) = \frac{\kappa(\|\mathbf{x}_i-\mathbf{x}_k\|_{\mathbf{w}})}{\sum_{j\in H_i}\kappa(\|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{w}})}, \forall k\in H_i \tag{11}$$

where $\kappa(\cdot)$ is the same kernel function as before. In particular, FSSun solves the following optimization:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \log\left(1+\exp\left(-\mathbf{w}^T\mathbf{z}_i\right)\right) + \lambda\|\mathbf{w}\|_1, \text{ s.t. } \mathbf{w}\ge 0 \tag{12}$$

The optimization in (12) can be reformulated to an unconstrained optimization by replacing w_l with w_l^2. Moreover, it should be noted that in FSSun a fixed-point recursion method, in which the derivative of the objective function with respect to $\mathbf{w}$ is obtained by considering $\mathbf{z}_i$ as a constant vector, is used to solve for $\mathbf{w}$ because $\mathbf{z}_i$ implicitly depends on $\mathbf{w}$ through the probabilities.

C. LMFW

LMFW [11] is a feature weighting algorithm based on KNN and largely inspired by the work of Weinberger et al. [15]. For easy of comparison with FSSun and NCFS, we describe LMFW under the Manhattan distance. The main idea of LMFW is that the k-nearest neighbors of each sample $\mathbf{x}_i$ should share the same label y_i while samples with different labels should be separated by a large margin. Before constructing the model of LMFW, the target neighbors of each sample $\mathbf{x}_i$, which are the samples with the same label y_i that we wish to be closest to $\mathbf{x}_i$, should be first determined. In the algorithm, the target neighbors are identified as the k nearest neighbors in the original feature space and do not change during the learning process. The notation $j\to i$ is used to indicate that sample $\mathbf{x}_j$ is a target neighbor of sample $\mathbf{x}_i$. In particular, the objective function of LMFW is given by:

$$\begin{aligned}&(1-\mu)\sum_{i=1}^{N}\sum_{j\to i} D_{\mathbf{w}}\left(\mathbf{x}_i,\mathbf{x}_j\right) + \mu\sum_{i=1}^{N}\sum_{j\to i}\sum_{k=1}^{N}(1-y_{ik})\left[1\right.\\ &\left.+D_{\mathbf{w}}\left(\mathbf{x}_i,\mathbf{x}_j\right) - D_{\mathbf{w}}\left(\mathbf{x}_i,\mathbf{x}_k\right)\right]_+ + \lambda\sum_{l=1}^{d} w_l^2\end{aligned} \tag{13}$$

where the hinge loss function $[z]_+ = \max(z,0)$, the indicator variable $y_{ik}=1$ if and only if $y_i = y_k$, and $y_{ik}=0$ otherwise, $\mu\in[0,1]$ is a balance parameter and $\lambda>0$ is a regularization parameter. It should be noted that the objective function has three competing terms, the first term penalizes large distances between each samples and its target neighbors, and the second term penalizes small distance between each sample and other samples with different labels, while the final term is expect to increase the sparseness of selected features. LMFW is implemented with Linear Programming in [11]. In fact, there exists a fast gradient descent method to minimize the objective function (13). For the detailed description, please refer to the literature [16].

IV. Experimental Results

Two different sets of experiments were carried out. In the first set, the NCFS algorithm was applied to well controlled, toy data in order to assess the ability of eliminating irrelevant features and to show the impact of different parameter settings. In the second, eight widely used microarray datasets were considered. The aim of these experiments was to illustrate the effectiveness of our algorithm in real application. Meanwhile, the comparison of NCFS with the other three algorithms FSSun, LMFW and Simba was made.

A. Toy data

In this section, our objective is to illustrate the behavior of the proposed method in the presence of the irrelevant features and to show the effects of different parameter

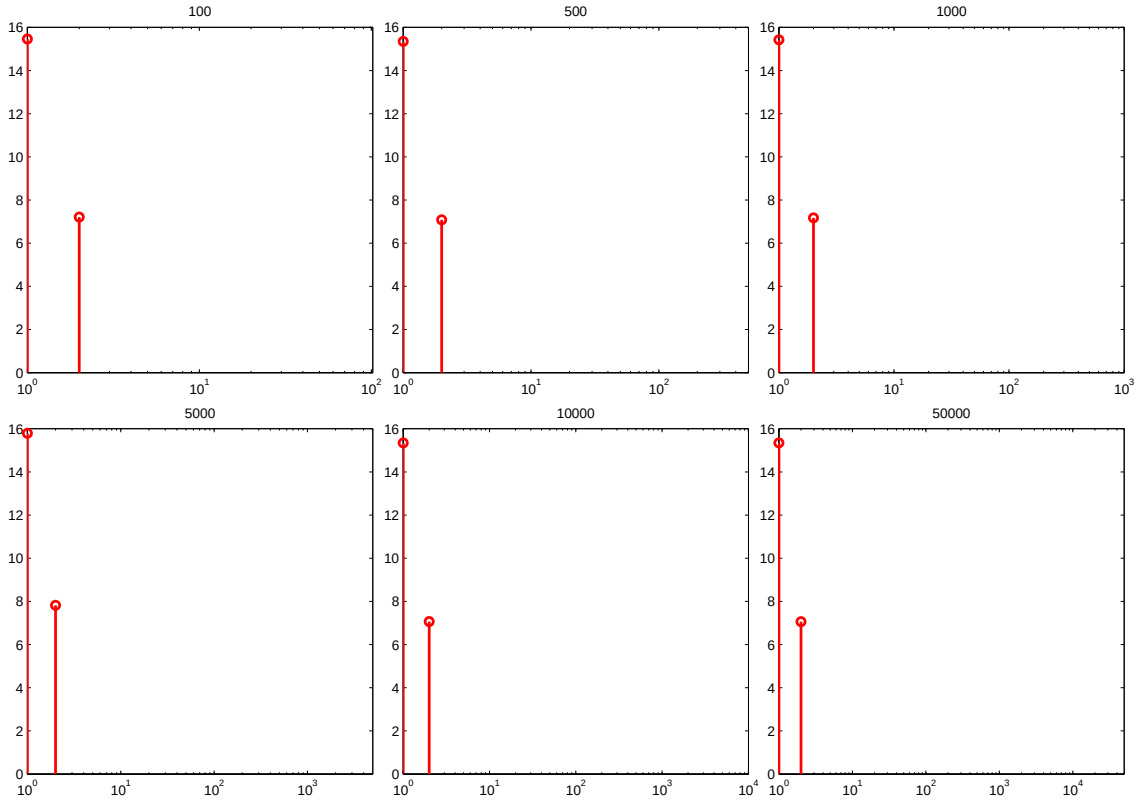


Figure 1. Feature weighting factors learned on the toy dataset with different numbers of irrelevant features, ranging from 100 to 50000. The horizontal coordinate represents the index of each feature, and the vertical coordinate denotes the corresponding weighting factors learned by the proposed algorithm. The two original features are always fixed in the first two indexes.

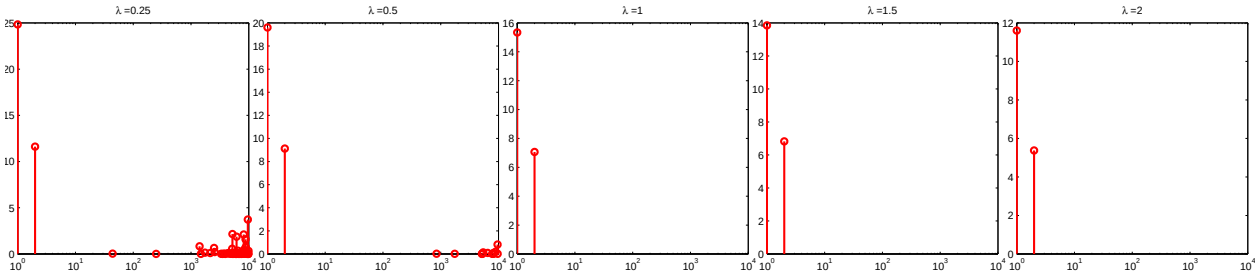


Figure 2. Feature weighting factors learned on the toy dataset with 10000 irrelevant features. The value of kernel width σ is set to 1, and regularization parameter λ is chosen from $\{0.25, 0.5, 1, 1.5, 2\}$.

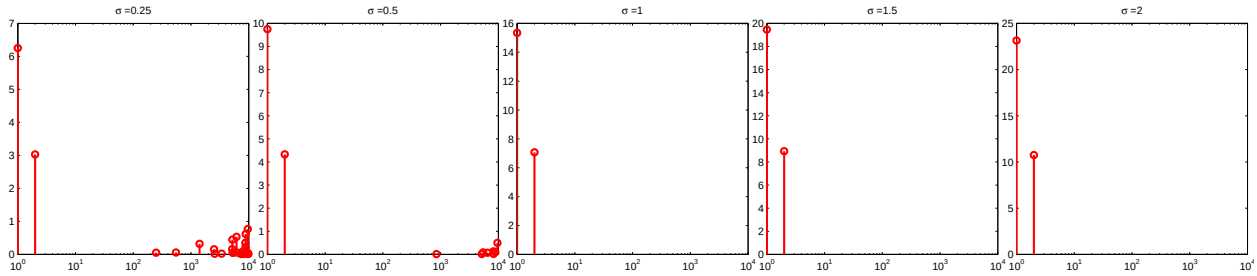


Figure 3. Feature weighting factors learned on the toy dataset with 10000 irrelevant features. The value of regularization parameter λ is set to 1, and kernel width σ is chosen from $\{0.25, 0.5, 1, 1.5, 2\}$.

settings. An artificial non-linear problem, previously considered in [11], [17], [18], was used. It involves two classes in two dimensions. The data for the first class are drawn from $N(\mu_1, \Sigma)$ or $N(\mu_2, \Sigma)$ with equal probability, where mean vectors $\mu_1 = \{-0.75, -3\}$, $\mu_2 = \{0.75, 3\}$ and covariance matrix $\Sigma = \mathbf{I}$. The data for the second class are drawn again from two normal distributions with equal probability, having mean vectors $\mu_1 = \{3, -3\}$,

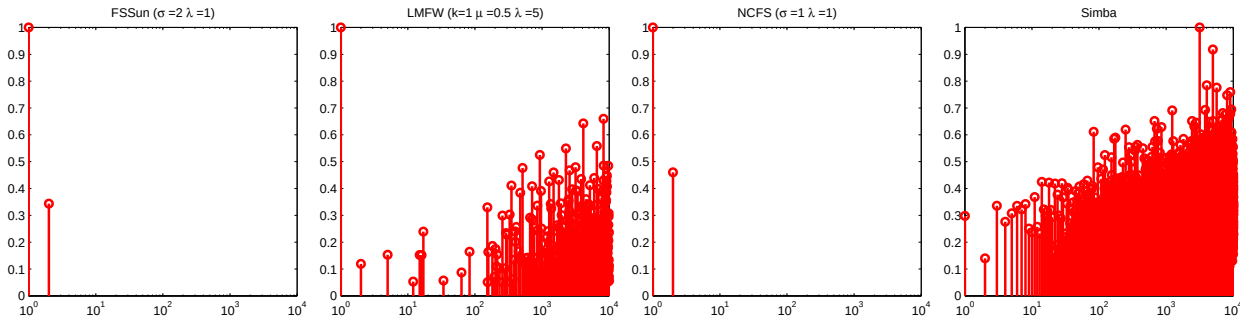


Figure 4. Normalized feature weighting factors of four algorithms learned on the toy dataset with 10000 irrelevant features.

$\mu_2 = \{-3, 3\}$ and the same Σ as before. The dataset with 200 samples is obtained by drawing 100 samples from each class independently. To assess the ability of the algorithm to filter irrelevant features, we added a varying number of irrelevant features to each sample. In this work, the added irrelevant features are drawn from a normal distribution with zero mean and variance 20 and the number of which is set to {100, 500, 1000, 5000, 10000, 50000}. All features are then normalized to the range between 0 and 1. Plots of the learned feature weighting factors of our algorithm on the toy dataset with different numbers of irrelevant features are presented in Fig.1. For the results reported here, the kernel width and regularization parameter of our algorithm are set to 1 and 1, respectively. From the figure, we can see that the proposed algorithm is insensitive to the increase of irrelevant features. For the significantly varying number of irrelevant features, our algorithm gives the almost identical feature weight factors for the two original features. The weights associated with the irrelevant feature are all nearly close to zero, which means that our method can filter the noise features effectively.

Like FSSun, there are a kernel width parameter σ and a regularization parameter λ needing to be adjusted in our algorithm. To study the influences of different parameter settings on the learned feature weighting factors, we performed a test on the toy dataset with 10000 irrelevant features by fixing one parameter while altering the other. The experimental results under different parameter settings are shown in Figs. 2 and 3. It can be observed that the proposed method always give the reasonable weighting factors although the resulting weights vary as the parameters change. Therefore, the ability of our algorithm performing feature selection is insensitive to a specific choice of the two parameters σ and λ, which is the same as FSSun.

In addition, we compare our algorithm with three other algorithms, including FSSun, LMFW and Simba, on the toy dataset with 10000 irrelevant features. The learned feature weights of four algorithms are plotted in Fig.4. For ease of comparison, the maximum value of the learned feature weights of each algorithm was normalized to 1. From the figure, we can see that only FSSun and NCFS could obtain sparse weight vector and successfully identify the two original features by giving them the largest weights. LMFW identifies one and Simba none. The parameter settings of three algorithms FSSun, LMFW and NCFS are also given in Fig.4. In fact, we test a wide range of parameter values of LMFW and found that the resulting weights have no significant difference. The poor performance of LMFW may be because its target neighbors are determined by nearest neighbors in original feature space at the outset of learning and do not change during the learning process. However, the nearest neighbors in original feature space may not be true in the weighted feature space, especially when the feature dimension is very high. Furthermore, the maintenance of unit margin damages the sparseness of its weight vector to some extent. For Simba algorithm, its matlab source code was downloaded from [8]. The sigmoid utility function with default parameter is used since it is less sensitive to outliers than the linear utility. The number of passes of training data and the number of starting points are set to 10 and 5, respectively. Due to the fact that the implementation of Simba is based on stochastic gradient ascent algorithm, we found that its solution vector is different in two runs with the same settings.

Moreover, it should be noted that if we perform feature selection by threshold, i.e., the features whose normalized weighting factor is bigger than a certain threshold are considered useful, the algorithms without sparse solution vector, such as LMFW and Simba, will choose more irrelevant features. Since these algorithms also have possibility to give the larger weight factors for important features, therefore, it is unfair to compare FSSun and NCFS with them based on the threshold. In fact, it is also difficult to determine a suitable threshold for FSSun and NCFS in practical application. Therefore, in the next section, we will perform comparative experiments by weight ranking for fair consideration.

B. Microarray datasets

To investigate the ability of the proposed method in practical applications, we further performed the experiments to compare it with three different algorithms LMFW, Simba and FSSun on the eight widely used microarray datasets [19], [20], [21], including Brain_Tumor2, Colon_Cancer, DLBCL, Leukemia1, Lung_Cancer, Lymphoma, Prostate_Tumor and SRBCT. For these datasets, which include 4 two class problems

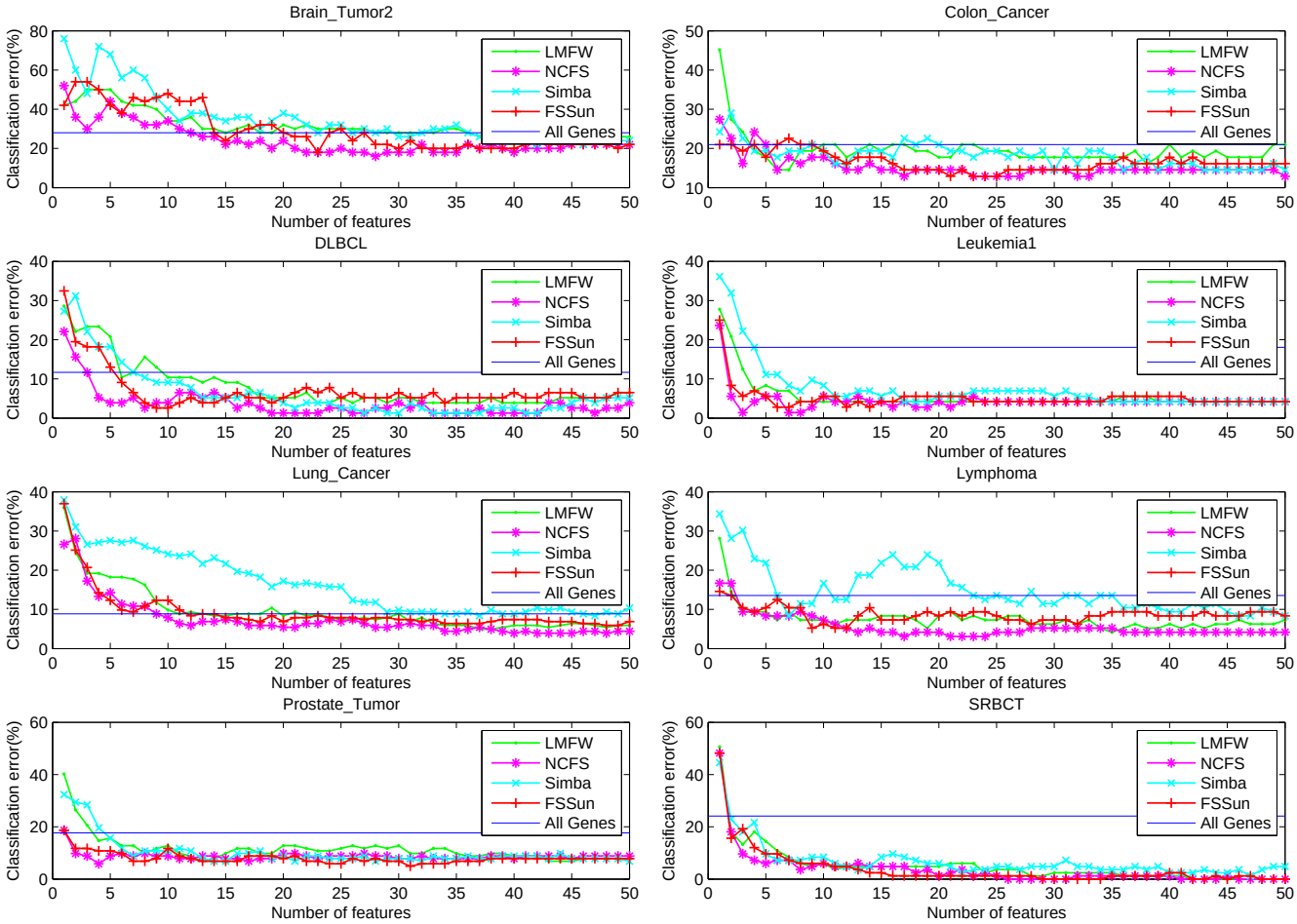


Figure 5. Comparison of four algorithms via the classification error on eight microarray datasets.

TABLE I.
MICROARRAY DATASETS USED IN THE EXPERIMENTS.

Datasets	Samples	Classes	Features
Brain_Tumor2	50	4	10367
Colon_Cancer	62	2	2000
DLBCL	77	2	5469
Leukemia1	72	3	5327
Lung_Cancer	203	5	12600
Lymphoma	96	2	4026
Prostate_Tumor	102	2	10509
SRBCT	83	4	2308

and 4 multi-class problems, one significant characteristic is that the number of samples is remarkably less than the number of features. The detailed data information is given in Table I. Note that the intention of microarray-based analysis is to mine a set of genes that show a strong relationship to phenotypes of interest. Since redundant genes will increase medical examination costs unnecessarily in clinical experiments, the number of resulting genes should be as small as possible. This means that a good feature selection method for microarray analysis should achieve a better prediction performance with the smaller number of features.

As in [10], [13], each feature variable in the raw data is normalized to $[0, 1]$. Since the sample number is very small, we used the leave-one-out cross validation method to carry out a test. Note that the four feature selection methods LMFW, Simba, FSSun and NCFS are all based on nearest neighbor model. Therefore, the KNN with Manhattan distance is used to evaluate the quality of selected features of each algorithm. The only parameter K for KNN is set to 3 for all the tests as in [13]. Before performing the leave-one-out tests, the parameters of four algorithms LMFW, Simba, FSSun and NCFS must be determined. For the first algorithm LMFW, the number of target neighbors is set to 3 and the parameters μ and λ were chosen from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\{0.1, 0.5, 1, 5, 10\}$ by cross validation, respectively. The settings of Simba are the same as before. For FSSun and NCFS, their kernel widths and regularization parameters are set to 5, 1 and 1, 1, respectively. It should be noted that the parameter settings of FSSun in this study are the same as in [13].

The curves of classification errors of KNN based on the 50 top ranked features derived from four feature selection algorithms on eight microarray datasets are plotted in Fig.5. In order to make a comparison, the classification errors of KNN using all genes are also reported. It can be seen that KNN obtain the significant performance

TABLE II.
CLASSIFICATION ERRORS OF FOUR ALGORITHMS ON EIGHT MICROARRAY DATASETS.

Datasets	LMFW	NCFS	Simba	FSSun
Brain_Tumor2	24.0(41)	**16.0**(28)	22.0(42)	18.0(23)
Colon_Cancer	14.52(6)	**12.9**(17)	14.52(30)	12.9(21)
DLBCL	3.9(23)	**1.3**(19)	**1.3**(27)	2.6(9)
Leukemia1	4.17(8)	**1.39**(3)	4.17(17)	2.78(6)
Lung_Cancer	4.93(37)	**3.94**(40)	8.37(37)	5.91(48)
Lymphoma	4.17(35)	**3.12**(17)	8.33(7)	5.21(9)
Prostate_Tumor	6.86(13)	5.88(4)	6.86(14)	**4.9**(31)
SRBCT	**0**(42)	**0**(26)	1.2(47)	**0**(29)
win/tie/loss	0/1/7	4/3/1	0/1/7	1/2/5

The number in the brackets represents the number of optimal features at which the minimum classification error is obtained. Bold number indicates the minimum classification error of each row. The last row records the win/tie/loss of each algorithm according to the classification error.

improvement by performing feature selection. Moreover, in Table II, we record the minimum classification error of four algorithms on the 50 top ranked features. The numbers of optimal features at which the minimum classification error is obtained are also presented in Table II. From the table, it can be observed that except for Prostate_Tumor, in which FSSun obtains the best classification result, for the remaining seven datasets, NCFS is the clear winner compared to LMFW, Simba and FSSun according to the classification error and the number of optimal features. One possible explanation is that the model of NCFS is more closely related to the leave-one-out classification error than that of the other three methods. We can also find that among the used feature selection methods, FSSun is most comparable to NCFS. Moreover, it is interesting to note that, on DLBCL dataset, Simba also achieves the minimum classification error although with more features compared to NCFS, whereas FSSun and LMFW don't. The win/tie/loss of four algorithms based on the classification error is also given in Table II.

V. CONCLUSION

In this paper, we present a novel feature weighting method in the context of NN. The proposed method, which is called NCFS, uses the gradient ascent technique to maximize the expected leave-one-out classification accuracy with a regularization term. The effectiveness of this algorithm has been evaluated through a number of experiments involving a toy data and eight microarray datasets. Meanwhile, the impact of two parameters, the kernel width σ and the regularization parameter λ, has been studied empirically. Overall, the proposed method is insensitive to a specific choice of the two parameters.

REFERENCES

[1] H. Liu, E. Dougherty, J. Dy, K. Torkkola, E. Tuv, H. Peng, C. Ding, F. Long, M. Berens, L. Parsons *et al.*, "Evolving feature selection," *IEEE Intelligent Systems*, vol. 20, no. 6, pp. 46–76, 2005.

[2] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, *Feature Extraction: Foundations and Applications*. Springer-Verlag, 2006.

[3] S. Maldonado, R. Weber, and J. Basak, "Simultaneous feature selection and classification using kernel-penalized support vector machines," *Information Sciences*, vol. 18, pp. 115–128, 2011.

[4] J. Miranda, R. Montoya, and R. Weber, "Linear penalization support vector machines for feature selection," *Pattern Recognition and Machine Intelligence*, pp. 188–192, 2005.

[5] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[6] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[7] K. Kira and L. Rendell, "A practical approach to feature selection," in *Proceedings of the ninth international workshop on Machine learning*, 1992, pp. 249–256.

[8] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection-theory and algorithms," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, pp. 43–50.

[9] A. Navot, L. Shpigelman, N. Tishby, and E. Vaadia, "Nearest neighbor based feature selection for regression and its application to neural activity," in *Advances in Neural Information Processing Systems 18*. MIT Press, 2006, pp. 995–1002.

[10] Y. Sun, "Iterative relief for feature weighting: Algorithms, theories, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1035–1051, 2007.

[11] B. Chen, H. Liu, J. Chai, and Z. Bao, "Large margin feature weighting method via linear programming," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1475–1488, 2009.

[12] Y. Li and B. Lu, "Feature selection based on loss-margin of nearest neighbor classification," *Pattern Recognition*, vol. 42, no. 9, pp. 1914–1921, 2009.

[13] Y. Sun, S. Todorovic, and S. Goodison, "Local learning based feature selection for high dimensional data analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 99, 2009.

[14] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2005, pp. 513–520.

[15] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems 18*. Cambridge, MA: MIT Press, 2006, pp. 1473–1480.

[16] K. Weinberger and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *The Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[17] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for svms," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 668–674.

[18] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131–159, 2002.

[19] A. Statnikov, C. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," *Bioinformatics*, vol. 21, no. 5, pp. 631–643, 2005.

[20] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and

normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745–6750, 1999.

[21] A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu *et al.*, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

Wei Yang is currently a PhD candidate in the Biocomputing Research Centre at the Harbin Institute of Technology. His interests include bioinformatics and pattern recognition.

Kuanquan Wang is a full professor and PhD supervisor with School of Computer Science and Technology at Harbin Institute of Technology. He is a senior member of IEEE, a senior member of China Computer Federation and a senior member of Chinese Society of Biomedical Engineering. His main research areas include image processing and pattern recognition, biometrics, biocomputing, virtual reality and visualization. So far, he has published over 200 papers and 6 books, got 10 patents, and won 1 second prize of National Teaching Achievement.

Wangmeng Zuo is an associate professor in School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. He received the PhD degree in computer application technology from Harbin Institute of Technology in 2007. From July to December 2004, from November 2005 to August 2006, and from July 2007 to February 2008, he was a research assistant in the Department of Computing, Hong Kong Polytechnic University. From August 2009 to February 2010, he was a visiting professor in Microsoft Research Asia. His research interests include sparse representation, biometrics, pattern recognition, and computer vision.

New Public-Key Cryptosystem Based on Two-Dimension DLP

Xiaoqiang Zhang
State Key Lab of Software Development Environment, Beihang University, Beijing, China
Email: grayqiang@163.com

Guiliang Zhu, Weiping Wang and Mengmeng Wang
North China University of Water Conservancy and Electric Power, Zhengzhou, China
Email: zgg500@126.com; wangweiping888@qq.com; 295612840@qq.com

Shilong Ma
State Key Lab of Software Development Environment, Beihang University, Beijing, China
Email: slma@nlsde.buaa.edu.cn

***Abstract*—The asymmetric cryptosystem plays an important role in the cryptology nowadays. It is widely used in the fields of data encryption, digital watermarking, digital signature, secure network protocol, etc. However, with the improvement of computing capability, longer and longer the key length is required to ensure the security of interaction information. To shorten the key length and improve the encryption efficiency, by defining the two-dimension discrete logarithm problem (DLP), a new public-key cryptosystem is proposed. This new cryptosystem generalizes the public-key cryptosystem from one dimension to two dimensions. The core algorithms of the proposed cryptosystem are also designed, including the fast algorithm, computing the inverse matrix modulo p and finding the period. To verify the correctness and rationality of the new cryptosystem, two examples are carried out. Meanwhile, the efficiency and security are analyzed in detail. Experimental results and theoretical analyses show that the new cryptosystem possesses the advantages of the outstanding robustness, short key length, high security and encrypting many data once.**

***Index Terms*—asymmetric cryptosystem, discrete logarithm problem (DLP), two dimensions, RSA, ECC (elliptic curve cryptosystem)**

I. INTRODUCTION

The cryptosystems can be classified as the symmetrical cryptosystem and the asymmetrical cryptosystem (also named as the public-key cryptosystem) by the characteristics of the key. For the symmetrical cryptosystem, $n(n-1)/2$ keys are required to satisfy the secure communication among n users over the Internet. The key distribution and management become very difficult when n is a very large number. However, the asymmetrical cryptosystem just requires $2n$ keys, whose key distribution and management are much easier. Meanwhile, the asymmetrical cryptosystem cannot only use for data encryption [1-3], but also for digital signature and authentication [4-6].

As a landmark of the cryptology development, W. Diffie and M. E. Hellman proposed the concept of public-key cryptosystem in 1976 [7]. Afterwards many public-key cryptosystems are proposed. Experts in cryptology select three types of asymmetrical cryptosystem, which are regarded as the secure and efficient cryptosystems. The detailed description is as follows.

- The cryptosystem based on the integer factorization problem (IFP): Its representative is RSA (Rivest, Shamir, Adleman), which was proposed in 1977 [8]. The advantages of RSA are the simple principle and easy application. RSA cryptosystem is designed based on two big prime numbers p and q instead of a two-dimension matrix. Meanwhile, the plaintext M is segmented to several data blocks m_i, $i = 1, 2, 3, \cdots$ in advance, and each data block m_i can correspond a decimal number n_i. RSA can only encrypt the number n_i instead of a two-dimension plaintext matrix during an encryption process [9]. However, with the improvement of the integer factorization algorithm, we need to continuously lengthen the key length of RSA to ensure the security of the cipher text. 768 bits RSA is unsecure at present, and experts suggest applying 1024 bits RSA to ensure the 10-year security. To ensure the 20-year security, we are required to choose 2048 bits RSA. Although the extending of the key length can enhance the security of RSA cryptosystem, the encryption speed reduces sharply and the application becomes very difficult.
- The cryptosystem based on the discrete logarithm problem (DLP): Its representative is DSA (digital

Manuscript received January 30, 2011; revised June 25, 2011; accepted June 26, 2011.
Corresponding author: Xiaoqiang Zhang.

doi:10.4304/jcp.7.1.169-178

signature algorithm), which was proposed by the National Institute of Standards and Technology (NIST) in August 1991. It is a United States Federal Government standard for digital signatures.

- The cryptosystem based on the elliptic curve discrete logarithm problem (ECDLP): Its representative is ECC (elliptic curve cryptosystem). Koblitz and Miller proposed ECC in 1985 [10, 11]. The ElGamal scheme of ECC is designed based on a basic point of the elliptic curve instead of a two-dimension matrix. Meanwhile, the plaintext M is encoded to several corresponding points of the elliptic curve in advance, and then ECC can only encrypt one of these points instead of a two-dimension plaintext matrix during an encryption process [12]. ECC possesses the advantages of the short key length, fast speed, etc [13]. There is not an effectively deciphered method at present [15].

With the improvement of computing capability, the key length becomes longer and longer to maintain the security of interaction information. The increasing of the key length makes the encryption efficiency low. Meanwhile, current asymmetric cryptosystems are one dimension, which can only encrypt a datum once. To improve the encryption efficiency, it is reasonable to generalize the public-key cryptosystem from one dimension to two dimensions, and even high dimensions. Therefore, by defining two-dimension DLP, a new public-key cryptosystem is proposed in this paper. This cryptosystem generalizes the public-key cryptosystem from one dimension to two dimensions.

The rest of the paper is organized as follows. We generalize the definition of DLP from one dimension to two dimensions in Section II. Section III designs the new cryptosystem based on the two-dimension DLP. The fast algorithm for the proposed cryptosystem, the algorithm of computing the inverse matrix modulo p and the algorithm of finding the period are designed in Section IV. To verify the correctness and rationality of the new cryptosystem, two examples are carried out respectively in Sections V and VI. The efficiency analysis of the new cryptosystem is given in Section VII. The security analysis of the new cryptosystem is discussed in Section VIII. Section IX concludes the paper.

II. DLP

Solving DLP is a difficult mathematic problem at present, which plays an important role in cryptology. DLP is described as follows.

Let G be an Abelian group comprised of numbers. $\langle\alpha\rangle$ denotes a subgroup of G generated by $\alpha \bmod p$, where $p>3$ and p is a prime. Supposing $\beta \in \langle\alpha\rangle$, the discrete logarithm $\log_\alpha \beta$ is the smallest non-negative integer x such that $\alpha^x \bmod p = \beta$. DLP is computing $x = \log_\alpha \beta$ under the premise of given α and β [14]. For easy discussion, we call this problem as one-dimension DLP.

The matrix possesses more elements and complexer construction than a number. As an extension, we define the two-dimension DLP as follows.

Let G be an Abelian group comprised of matrixes. $\langle A\rangle$ denotes the subgroup of G generated by a matrix $A \bmod p$, where $p>3$ and p is a prime. Supposing $B \in \langle A\rangle$, the discrete logarithm is the smallest non-negative integer x such that $A^x \bmod p = B$. The two-dimension DLP is computing $x = \log_A B$ under the premise of given A and B. Notice that $\log_A B$ is just a denotation to unify the form.

III. NEW PUBLIC-KEY CRYPTOSYSTEM

A new public-key cryptosystem based on the two-dimension DLP is designed by using slightly altered Elgamal cryptosystem [15]. Supposing the following scenario, Alice is a sender and Bob is a receptionist. The detailed steps of this new cryptosystem are described as follows.

A. Bob's key-generating steps

Bob needs to generate his public key and private key before decrypting Alice's information. The detailed steps are as follows.

- Choose a matrix $A_{n\times n}$ comprised of the elements from the set $Z_p = \{0,1,2,\cdots,p-1\}$, where $|A| \neq 0$, A cannot be the identity matrix I, p is a prime or $p = 2^m$, and $\gcd(|A|, p) = 1$.
- Compute the period T of the generator $A \bmod p$, and then he can obtain a cyclic group $G_p = \{I, A, A^2, A^3, \cdots, A^{T-1}\}$.
- Randomly select an integer $d \in \{1,2,\cdots,T-1\}$ as his private key and calculate the matrix $Q = A^d \bmod p$.
- Publish the public key $[A, T, p, Q]$.

B. Alice's encryption steps

Alice performs the following steps to encrypt the plaintext.

- Acquire Bob's public key $[A, T, p, Q]$.
- After randomly selecting an integer $u \in \{1,2,\cdots, T-1\}$, Alice calculates the matrixes $C = A^u \bmod p$ and $D = Q^u \bmod p$ with Bob's public key.
- Let a matrix $M_{n\times n}$ be the plaintext, whose elements are from the set Z_p. Calculate the cipher text matrix $E = (D \times M) \bmod p$.
- Send the encrypted data $[C, E]$ to Bob.

C. Bob's decryption steps

Bob performs the following steps after receiving Alice's encrypted data $[C, E]$.

- Calculate the matrix $D = C^d \bmod p$ with his private key d. $D = Q^u = (A^d)^u = (A^u)^d = C^d \bmod p$.
- Compute the inverse matrix of D with the equation $(D \times D_p^{-1}) \bmod p = I$, where I is the identity matrix.
- Bob can recover the plaintext M by calculating $M = (D^{-1} \times E) \bmod p$.

IV. ALGORITHMS

A. Fast algorithm

$A^e \bmod p$ is the core operation for the proposed cryptosystem. To improve the encryption efficiency, the square-multiply method for numbers is generalized to matrixes.

The square-multiply method is one of the rapidest methods of computing modular exponentiation [16-19]. The algorithm idea derives from "QinJiuShao" algorithm, which was proposed in Song Dynasty of China [16]. To accelerate the computing, the square-multiply method is used in RSA [16, 18] and ECC [20, 21] at present.

The principle of the square-multiply method is converting the exponent into its binary expression at first. According to the binary expression, the result can be obtained conveniently by changing the expression of modular exponentiation.

E.g., first, convert the exponent e into a binary expression $(e_r, e_{r-1}, \cdots, e_1, e_0)_B$ to compute $a^e \bmod p$, where $e_r = 1$, $e_i \in \{0,1\}$, $i = r-1, \cdots, 1, 0$ and $r = \lfloor \log_2 a \rfloor$. Second, the expression of $a^e \bmod p$ is changed by using Equation (1), and then we can compute the result of $a^e \bmod p$ rapidly.

$$a^e \bmod p = a^{e_r \times 2^r + \cdots + e_1 \times 2^1 + e_0 \times 2^0} \bmod p = a^{e_r \times 2^r} \times \cdots \times a^{e_1 \times 2^1} \times a^{e_0 \times 2^0} \bmod p = ((((\cdots (a^{e_r})^2 \bmod p \times \cdots \bmod p)^2 \times a^{e_1}) \bmod p)^2 \times a^{e_0}) \bmod p. \quad (1)$$

The computing complexity of the square-multiply method is due to the length of the binary expression and the number of "1" in the binary expression.

Similarly, if the number a is substituted for a matrix A, Equation (1) can be rewritten as follows.

$$A^e \bmod p = A^{e_r \times 2^r + \cdots + e_1 \times 2^1 + e_0 \times 2^0} \bmod p = A^{e_r \times 2^r} \times \cdots \times A^{e_1 \times 2^1} \times A^{e_0 \times 2^0} \bmod p = ((((\cdots (A^{e_r})^2 \bmod p \times \cdots \bmod p)^2 \times A^{e_1}) \bmod p)^2 \times A^{e_0}) \bmod p. \quad (2)$$

Therefore, we can rapidly compute $A^e \bmod p$ with the square-multiply method.

B. How to compute the inverse matrix modulo p

The inverse matrix modulo p of the square matrix A is the matrix A_p^{-1} satisfying $(A \times A_p^{-1}) \bmod p = I$, where I is the identity matrix. Meanwhile, A and A_p^{-1} are made up of the elements from the set Z_p. Notice that only if $\gcd(|A|, p) \neq 1$, A_p^{-1} is existing. The process of computing the inverse matrix modulo p is as follows.

- According to $\gcd(|A|, p) = 1$, judge whether A_p^{-1} exist or not. If $\gcd(|A|, p) = 1$, A_p^{-1} exist. Otherwise, A_p^{-1} does not exist.
- If A_p^{-1} exist, compute $|A|$. Otherwise, end.
- Find $i \in Z_p$ satisfying $(|A| \times i) \bmod p = 1$.
- Compute $A_p^{-1} = (i \times |A| \times A^{-1}) \bmod p$, where A^{-1} is the inverse matrix in general.

The Matlab code for computing A_p^{-1} is given as follows.

```
function inverse=invmod(A, p)
%Compute the inverse matrix modulo p of the square matrix A.
%A and its inverse matrix modulo p are made up of the elements from the set Zp={0, 1, 2, ... , p-1}.
%Check input
[row,col]=size(A);
if row~=col
   error('The first parameter must be a square matrix.');
end
det_A=det(A);
%Judge whether the inverse matrix modulo p exist
if gcd(det_A, p)~=1
   fprintf('The inverse matrix modulo %d does not exist.\n', p);
   return
end
%Compute the inverse matrix modulo p
for i=1:p
   if mod(mod(det_A,p)*i, p)==1
      inv_det_A=i;
   end
end
inverse=mod(round(inv_det_A*det_A*inv(A)),p);
```

C. How to find the period T

Taking a square matrix $A_{n \times n}$ as the generator, a cyclic group $G_p = \{I, A, A^2, A^3, \cdots, A^{T-1}\}$ can be obtained, where p is a prime or $p = 2^m$. To find the period T, we design the steps of the algorithm as follows.

- Compute the elements in the first column of $B(:,1) = A \times A(:,1)$.
- Compute $B(:,1) = A \times B(:,1)$ repeatedly, until $B(1,1) = I(1,1)$, where I is the identity matrix.
- Supposing the times of repeated multiplying A is i now, we compute $C = A^i \bmod p$ with the fast algorithm offered in Section IV(A).
- Judge whether $C == I$ or not. If C is equal to I, then $T = i$; else go to Step 2.

The pseudocode for computing the period T is as follows.

```
i=1; B=A; C=A;
While C==I //I is the identity matrix.
{   While B(1, 1)== I(1, 1) //Only judge the first element.
{       B(:, 1)=(A*B(:, 1)) mod p; //Only compute elements in the first column.
i=i+1;
}
C=A^i mod p; //compute A^i with the fast algorithm offered in Section IV(A).
}
T = i;
```

V. A SMALL EXAMPLE

To verify the correctness and rationality of the proposed cryptosystem, a small example is carried out with a matrix $A_{3\times3}$. The result indicates that Bob can recover successfully the plaintext with the new cryptosystem. The detailed description is given as follows.

A. Bob's key-generating steps

- Choose the matrix A and the prime $p = 199867$.

$$A = \begin{bmatrix} 35229 & 81087 & 186969 \\ 183258 & 81999 & 178611 \\ 11570 & 70526 & 162525 \end{bmatrix}.$$

- Compute the period $T = 36987216$ of $A \bmod 199867$.
- Randomly choose the private key $d = 97131$ and calculate the matrix

$$Q = A^{97131} \bmod 199867 = \begin{bmatrix} 146146 & 303 & 187134 \\ 97027 & 71586 & 196024 \\ 58367 & 115209 & 91566 \end{bmatrix}.$$

- Publish the public key $[A, 36987216, 97131, Q]$.

B. Alice's encryption steps

- Acquire Bob's public key $[A, 36987216, 97131, Q]$.
- After selecting an integer $u = 3925$, Alice calculates the matrixes C and D with Bob's public key,

$$C = A^{3925} \bmod 199867 = \begin{bmatrix} 185342 & 188610 & 107335 \\ 147092 & 59828 & 86685 \\ 61984 & 128955 & 156400 \end{bmatrix},$$

$$D = Q^{3925} \bmod 199867 = \begin{bmatrix} 158335 & 135371 & 118290 \\ 180294 & 148209 & 128784 \\ 175149 & 105464 & 125418 \end{bmatrix}.$$

- Let the matrix M be the plaintext. She can calculate the cipher text matrix E.

$$M = \begin{bmatrix} 136164 & 75845 & 166248 \\ 100495 & 141799 & 85721 \\ 60882 & 37905 & 38660 \end{bmatrix},$$

$$E = (D \times M) \bmod 199867 = \begin{bmatrix} 158976 & 121301 & 187224 \\ 108166 & 176611 & 42960 \\ 95740 & 189640 & 129183 \end{bmatrix}.$$

- Send the encrypted data $[C, E]$ to Bob.

C. Bob's decryption steps

Bob performs the following steps after receiving Alice's encrypted data $[C, E]$.

- Calculate the matrix D with his private key $d = 97131$.

$$D = C^{97131} \bmod 199867 = \begin{bmatrix} 158335 & 135371 & 118290 \\ 180294 & 148209 & 128784 \\ 175149 & 105464 & 125418 \end{bmatrix}.$$

- Compute the matrix

$$D_{199867}^{-1} = \begin{bmatrix} 5668 & 103764 & 100957 \\ 19960 & 146800 & 40609 \\ 75844 & 105348 & 165025 \end{bmatrix}.$$

- Bob recovers the plaintext, $M = (D_{199867}^{-1} \times E)$

$$\bmod 199867 = \begin{bmatrix} 136164 & 75845 & 166248 \\ 100495 & 141799 & 85721 \\ 60882 & 37905 & 38660 \end{bmatrix}.$$

VI. ANOTHER EXAMPLE

To verify the correctness and rationality of the proposed cryptosystem, another example is carried out using the digital image Lena. The detailed description is given as follows.

A. Experimental steps

(1) Bob's key-generating steps

- Choose the digital image Tiffany as the matrix $A_{512\times512}$ and the prime $p = 256$, as shown in Fig.1.

Figure 1. Tiffany

- Because there are 512×512 elements in $A_{512\times512}$ and the element $a_{i,j} \in \{0, 1, 2, \cdots, 255\}$, the period T of $A \bmod 256$ is a big-integer and $T \le 2^{8\times512\times512}$.
- Randomly choose the private key $d = 44536346\ 54354354543543543454 \approx 2^{92}$ and calculate the matrix $Q = A^d \bmod 256$. The result of Q is as shown in Fig. 2.

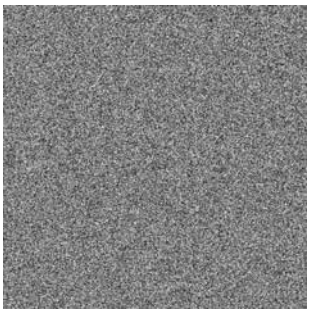


Figure 2. Q matrix

- Publish the public key $[A,T,p,Q]$.

(2) Alice's encryption steps

- Acquire Bob's public key $[A,T,p,Q]$.
- After randomly selecting an integer $u = 57435435\ 43543543345799853845$, Alice calculates the matrixes $C = A^u \mod 256$ and $D = Q^u \bmod 256$ with Bob's public key. The results of C, D are respectively as shown in Fig. 3 and Fig. 4.

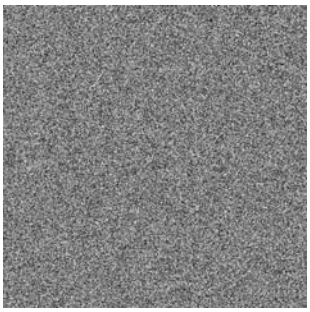


Figure 3. C matrix

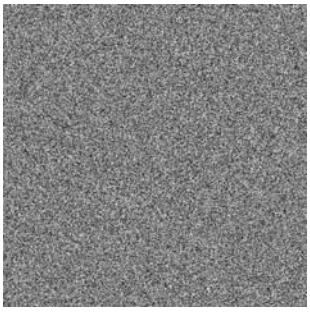


Figure 4. D matrix

- Let the digital image Lena be the plaintext $M_{512\times512}$, as shown in Fig. 5. She can calculate the cipher text matrix $E = (D \times M) \bmod 256$. E is the encrypted image, as shown in Fig. 6.
- Send the encrypted data $[C,E]$ to Bob.

(3) Bob's decryption steps

Bob performs the following steps after receiving Alice's encrypted data $[C,E]$.

Figure 5. Lena

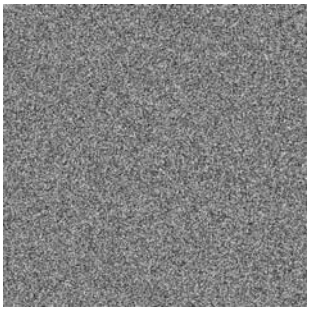


Figure 6. Encrypted image

- Calculate the matrix $D = C^d \bmod 256$ with his private key $d = 44536346543543545435435434 5 \approx 2^{92}$.
- Compute the matrix D_p^{-1}, and the result is as shown in Fig. 7.

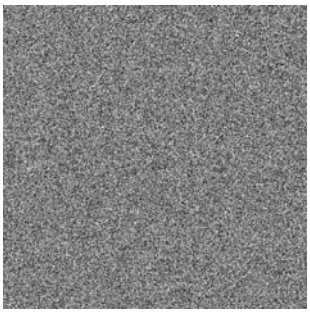


Figure 7. The inverse matrix of D

- Bob recovers the original image by computing $M = (D_p^{-1} \times E) \bmod 256$, and the result is as shown in Fig. 8.

Figure 8. Decrypted image

B. Robustness analysis

The correlation coefficient between two images with the same size is defined as follows [22],

$$\rho_{XY} = \frac{\mathrm{cov}(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}, \quad (3)$$

where X and Y are gray images, $E(X) = \frac{1}{N}\sum_{i=1}^{N} x_i$, $\mathrm{cov}(X,Y) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(X)][y_i - E(Y)]$, $D(X) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(X)]^2$, N is the pixel number of the images X and Y, $x_i \in X$ and $y_i \in Y$ are two pixels in the corresponding position.

NPCR (Number of Pixels Change Rate) between two images with the same size is defined as follows [23],

$$f(i,j) = \begin{cases} 0 & X(i,j) = Y(i,j) \\ 1 & X(i,j) \neq Y(i,j) \end{cases}, \quad (4)$$

$$NPCR = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)}{m \times n} \times 100\%, \quad (5)$$

where $X_{m\times n}$ and $Y_{m\times n}$ are images.

UACI (Unified Average Changing Intensity) between two images with the same size is defined as follows [23],

$$UACI = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} |X(i,j) - Y(i,j)|}{255 \times m \times n} \times 100\%, \quad (6)$$

where $X_{m\times n}$ and $Y_{m\times n}$ are images.

With the criterions of correlation coefficient, NPCR and UACI, we analyze the robustness of the proposed cryptosystem as follows.

(1) Without noise effect

Supposing that the encrypted image isn't damaged during the storage or transmission, the decrypted image is the same as the original image. Therefore, for the decrypted image and original image, the correlation coefficient is ρ_{XY} =1, $NPCR$ =0 and $UACI$ =0.

(2) Salt and peppers noise attack

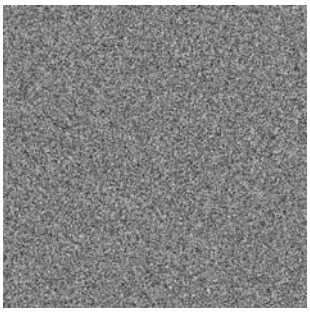


Figure 9. Salt and peppers noise (0.03)

To simulate the noise channel, we add the salt and peppers noise (0.03) to the encrypted image, as shown in Fig. 9. The corresponding decrypted image is as shown in Fig. 10. Finally, for the decrypted image and original image, the correlation coefficient is ρ_{XY} = 0.8855, $NPCR$ =9.841% and $UACI$ = 7.91%. Therefore, the new cryptosystem are robust against the salt and peppers noise attack.

Figure 10. Decrypted image

(3) Cutting attack

To simulate the tampering operation during the storage or transmission, we cut the encrypted image (6.25%), and the tampered image is as shown in Fig. 11. The corresponding decrypted image is as shown in Fig. 12. Finally, for the decrypted image and original image, the correlation coefficient is ρ_{XY} = 0.8877, $NPCR$ =12.45% and $UACI$ = 2.84%. Therefore, the new cryptosystem are robust against the cutting attack.

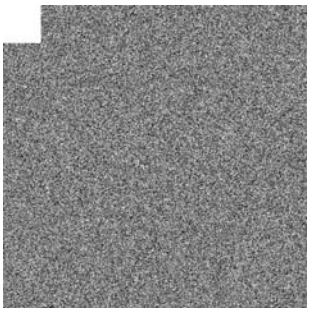


Figure 11. Cutting (6.25%)

Figure 12. Decrypted image

C. Encryption efficiency analysis

(1) Encryption efficiency analysis of ECC

The bottom-layer operations of ECC are the big-integer operations, such as big-integer addition, big-integer subtraction, big-integer multiplication and big-integer division. The core operation of ECC is the scalar multiplication operation. People have proposed some fast algorithms, such as the binary method, non-adjacent form (NAF) method and sliding window method. We realize the ElGamal scheme of ECC [15] with the binary method.

The algorithms were programmed in MyEclipse 7.5 on a PC with the Intel (R) Pen 4, CPU frequency 2.80 Ghz and Memory 2 GB.

Choose the 192 bits elliptic curve, which is recommended by NIST. The parameters of this elliptic curve are $a=-3$, $b=$ 245515554600894381774029391519745178476910805816119123806 5, and $p=$ 6277101735386680763835789423176059013767194773182842284081. The base point is $N=$ (602046282375688656758213480587526111916698976636884684818, 174050332293622031404857552280219410364023488927386650641), the period of N is $T=$ 6277101735386680763835789423176059013767194773182842284081. The private key is $d=$ 1000072627282998656287642690503854234 53895943703. To encrypt a pixel value with ECC, the encrypted time is about 42.1637 ms $\approx$ 42 ms. Therefore, to encrypt Lena $A_{512\times512}$ with ECC, the total encrypted time is about $42\times512\times512\approx11010$ s ≈3.0583 h.

(2) Encryption efficiency analysis of the proposed cryptosystem

The core operations of the proposed cryptosystem are multiplication of matrices and module operations, and we offer a fast algorithm in Section IV(A). The algorithms were programmed in Matlab 6.5 on a PC with the Intel (R) Pen 4, CPU frequency 2.80 Ghz and Memory 2 GB. The encrypted time in experiment is 43.0531 s. Therefore, the proposed cryptosystem is efficient to satisfy the requirement in the practical application.

VII. Efficiency Analysis

By comparing the difference between one-dimension DLP and two-dimension DLP, we analyze the temporal complexity of the new cryptosystem in detail as follows.

A. One-dimension DLP

Let $\langle\alpha\rangle$ denotes the group generated by a number $\alpha \bmod p$, where p is a prime or $p=2^m$. Suppose the order of $\langle\alpha\rangle$ is T_1. If Oscar (the attacker) adopts the brute-force attack, the worst case is that he should compute $\alpha^2=(\alpha\times\alpha) \bmod p$, $\alpha^3=(\alpha^2\times\alpha) \bmod p$, $\cdots$, $\alpha^{T_1}=(\alpha^{T_1-1}\times\alpha) \bmod p$. Under the premise of known the value of α^k, he only needs one time of multiplication operation ($\alpha^k\times\alpha$) and one time of modular operation ($\alpha^{k+1} \bmod p$) to obtain the value of $a^{k+1}=(\alpha^k\times\alpha) \bmod p$.

B. Two-dimension DLP

Let $\langle A\rangle$ denotes the group generated by a matrix $A_{n\times n} \bmod p$, where p is a prime or $p=2^m$. Supposed the order of $\langle A\rangle$ is T_2. If Oscar adopts the brute-force attack, the worst case is that he should compute $A^2=(A\times A) \bmod p$, $A^3=(A^2\times A) \bmod p$, $\cdots$, $A^{T_2}=(A^{T_2-1}\times A) \bmod p$. Under the premise of known the value of A^k, he needs to compute the elements of A^{k+1} at first to obtain the result of $A^{k+1}=(A^k\times A) \bmod p$. According to the matrix theory, we have $a_{ij}^{k+1}=a_{i1}^k\times a_{1j}+a_{i2}^k\times a_{2j}+\cdots+a_{in}^k\times a_{nj}$, where $a_{ij}^{k+1}\in A^{k+1}$, $a_{ij}^k\in A^k$ and $a_{ij}\in A$. Therefore, to obtain the value of the element a_{ij}^{k+1}, we need n times of multiplication operation, $n-1$ times of addition operation and one time of modular operation ($a_{ij}^{k+1} \bmod p$). In total, under the premise of known the value of A^k, to obtain the result of $A^{k+1}=(A^k\times A) \bmod p$, he needs $n^2\times n=n^3$ times of multiplication operation, $n^2(n-1)$ times of addition operation and n^2 times of module operation.

From the above analysis, we can see that solving the two-dimension DLP is more different than the one-dimension DLP. The temporal complexity of two-dimension DLP is n^2 times at least of one-dimension DLP. Therefore, for the same security level, the key length of the proposed cryptosystem is $1/n^2$ of the cryptosystems based on the one-dimension DLP in theory. In this sense, the key length of the new cryptosystem is shorter than the public-key cryptosystem based on the one-dimension DLP, such as Diffie-Hellman key exchange [7], Elgamal [24] and Massey-Omura [25].

To verify the above conclusion, several experiments are also performed. Let the generator of the group $\langle A\rangle$ be

$$A=\begin{bmatrix}4&7&1\\6&4&8\\9&6&4\end{bmatrix}.$$

We calculated the orders T of $\langle A\rangle$ for $p=163$, $p=1987$ and $p=199867$ respectively, as shown in Tab.1. Supposing that Oscar adopts the brute-force attack, the time needed for the worst case to attack one-dimension DLP and two-dimension DLP is shown in Tab. 1.

TABLE 1.
COMPARISON ON THE SPEED FOR ATTACKING ONE-DIMENSION DLP AND TWO-DIMENSION DLP.

p	T	Time for one-dimension DLP (s)	Time for two-dimension DLP (s)
163	$26568\approx2^{15}$	0.0047	0.0734
1987	$3948168\approx2^{22}$	0.5938	11.3594
199867	$36987216\approx2^{25}$	5.5620	107.1880

VIII. Security Analysis

A. Possible attack analysis

The security of the new public-key cryptosystem is based on the hardness of solving the two-dimension DLP. To illuminate the cryptosystem security, we consider the following three possible attacks.

(1) Attack 1

- Attack: Oscar tries to obtain Bob's private key d from Bob's public key $[A,T,p,Q]$.

- Attack analysis: From the principle of the public-key cryptosystem, Bob publishes his public key $[A,T,p,Q]$. Therefore, Oscar is available to obtain $[A,T,p,Q]$. From the steps of the new cryptosystem, we have $Q = A^d \bmod p$. Oscar is required to calculate $d = \log_A Q$. Therefore, obtaining Bob's private key d is equivalent to solving the two-dimension DLP.

(2) Attack 2

- Attack: Supposing that Oscar has obtained Bob's public key $[A,T,p,Q]$ and the cipher text $[C,E]$, he tries to compute Bob's private key d.
- Attack analysis: From the principle of the public-key cryptosystem, Oscar is available to obtain $[A,T,p,Q]$. We suppose that Oscar can obtain the cipher text $[C,E]$ fortunately. From the steps of the new cryptosystem, we have $C = A^u \bmod p$, i.e., $u = \log_A C$. He can calculate $D = Q^u \bmod p = C^d \bmod p$ with Bob's public key Q. Finally, he can compute $d = \log_C D$. During the whole process, Oscar should compute $u = \log_A C$ and $d = \log_C D$. Therefore, to obtain Bob's private key d, unless Oscar can solve the two-dimension DLP.

(3) Attack 3

- Attack: Supposing that Oscar has obtained Bob's public key $[A,T,p,Q]$ and the cipher text $[C,E]$, he tries to gain Alice's plaintext M.
- Attack analysis: From the principle of the public-key cryptosystem, Oscar is available to obtain $[A,T,p,Q]$. We suppose that Oscar can obtain the cipher text $[C,E]$ fortunately. From the steps of the new cryptosystem, we have $C = A^u \bmod p$, i.e., $u = \log_A C$. He can calculate $D = Q^u \bmod p$ with Bob's public key Q. Then he can compute D^{-1}. Finally, he can compute $M = (D^{-1} \times E) \bmod p$ with the plaintext E. Oscar needs to compute $u = \log_A C$ during the attack process. Therefore, to obtain Alice's plaintext M, unless the two-dimension DLP is solved.

The above analysis demonstrates that the difficulty of breaking our cryptosystem is equivalent to breaking the Elgamal cryptosystem and solving the two-dimension DLP.

B. Key space analysis

The key space is a set comprised of all the possible keys. For a secure image encryption algorithm, the key space should be large enough to make the brute force attack infeasible [26].

(1) Key space analysis of the cryptosystem based on the one-dimension DLP

Because the integer a are chosen from the set $Z_p = \{0,1,2,\cdots,p-1\}$ in the cryptosystems based on the one-dimension DLP, there are p possible cases of a. Therefore, the period T of the generator $a \bmod p$ is an integer which is not more than p. Since the private key d is randomly chosen from the set $\{1,2,\cdots,T-1\}$, the size of the key space is not over p.

(2) Key space analysis of the proposed cryptosystem

Because the elements of the matrix $A_{n\times n}$ are from the set $Z_p = \{0,1,2,\cdots,p-1\}$ in the new cryptosystem, there are $p^{n\times n}$ possible cases of $A_{n\times n}$. Therefore, the period T of the generator $A_{n\times n} \bmod p$ is an integer which is not over $p^{n\times n}$. Since the private key d is randomly chosen from the set $\{1,2,\cdots,T-1\}$, the size of the key space is not more than $p^{n\times n}$. For example, in the second example, the size of the key space is a big integer and not more than $2^{8\times 512\times 512}$.

In this sense, the key space $p^{n\times n}$ of the new cryptosystem is larger than the key space p of the public-key cryptosystem based on the one-dimension DLP. Therefore, the proposed cryptosystem possesses the advantage of high security.

IX. CONCLUSIONS

The definition of DLP is generalized from one dimension to two dimensions, and then a new public-key cryptosystem based on the two-dimension DLP is proposed, which generalizes the public-key cryptosystem from one dimension to two dimensions. The core algorithms of the new cryptosystem are offered, including fast algorithm, the algorithm of computing the inverse matrix modulo p, and the algorithm of finding the period T. Especially, to improve the efficiency, the square-multiply method for numbers is generalizing to matrixes. The theory analysis and experimental data show that the proposed cryptosystem possesses the advantages of the outstanding robustness, short key length, high security and encrypting many data once.

ACKNOWLEDGEMENTS

The authors like to express their sincere thanks to the anonymous reviewers and the editor P. Mahanti of Journal of Computers for their constructive comments and suggestions.

The authors thank Professor Xinqing Yan who works in North China University of Water Conservancy and Electric Power, for his careful reviews and valued suggestions. The authors also thank Ph.D. Jie Gai from Cameroon for pointing out several grammar mistakes.

REFERENCES

[1] Andrzej Chmielowiec, "Fixed points of the RSA encryption algorithm," *Theoretical Computer Science*, vol. 411, pp. 288–292, January 2010.

[2] Wongoo Lee, and Jaekwang Lee, "Design and implementation of secure e-mail system using elliptic curve cryptosystem," *Future Generation Computer Systems*, vol. 20, pp. 315–326, February 2004.
[3] Tzer-Shyong Chen, Kuo-Hsuan Huang, and Yu-Fang Chung, "A practical authenticated encryption scheme based on the elliptic curve cryptosystem," *Computer Standards & Interfaces*, vol. 26, pp. 461–469, May 2004.
[4] Coron JS, Naccache D, and Desmedt Y, "Index calculation attacks on RSA signature and encryption," *Designs, Codes and Cryptography*, vol. 38, pp. 41–53, January 2006.
[5] Tzer-Shyong Chen, "A specifiable verifier group-oriented threshold signature scheme based on the elliptic curve cryptosystem," *Computer Standards & Interfaces*, vol. 27, pp. 33–38, January 2004.
[6] Chen TS, Chung YF, and Huang GS, "Efficient proxy multisignature schemes based on the elliptic curve cryptosystem," *Computers & Security*, vol. 22, pp. 527–534, June 2003.
[7] Diffie, W., Hellman, M, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644–654, June 1976.
[8] Rivest, R., Shamir, and A., Aldeman, L, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, February 1978.
[9] Hung-Min Sun, Mu-En Wu, M. Jason Hinek, Cheng-Ta Yang, and Vincent S. Tseng, "Trading decryption for speeding encryption in rebalanced-RSA," *The Journal of Systems and Software*, vol. 82, pp. 1503–1512, September 2009.
[10] V. Miller, "Uses of elliptic curves in cryptography," *Advances in Cryptology-Crypto'85*, Berlin: Springer-Verlag, pp. 417–426, 1985.
[11] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 203–209, July 1987.
[12] Fu Minfeng, and Chen Wei, "Elliptic curve cryptosystem ElGamal encryption and transmission scheme," *International Conference on Computer Application and System Modeling*, vol. 6, pp. 51–53, June 2010.
[13] Tzer-Shyong Chen, "A threshold signature scheme based on the elliptic curve cryptosystem," *Applied Mathematics and Computation*, vol. 162, pp. 1119–1134, March 2005.
[14] J.A. Muir, D.R. Stinson, "On the low hamming weight discrete logarithm problem for nonadjacent representations," *Applicable Algebra in Engineering, Communication and Computing*, vol. 16, pp. 461–472, June 2006.
[15] Guiliang Zhu, and Xiaoqiang Zhang, "Mixed image element encryption algorithm based on an elliptic curve cryptosystem," *Journal of Electronic Imaging*, vol. 17, pp. 023007.1–5, February 2008.
[16] Dong Fuguo, and Li Yurong, "Study on QinJiuShao algorithm and its application in RSA," *Computer Engineering and Applications*, vol. 44, pp. 65–78, June 2008. (in Chinese)
[17] Adleman L M, "Factoring numbers using singular integers," *The 23rd Annual ACM Symposium on the Theory of Computing*, New Orleans, pp. 64–71, 1991.
[18] Li Qiang, and Zhang Jiyong, "A new fast RSA algorithm," *Mini-Micro Systems*, vol. 22, pp. 70–72, January 2001.
[19] Bergeron F, J. Berstel, S. Brlek, and C. Duboc, "Addition chains using continued fractions," *Journal of Algorithms*, vol. 10, pp. 403–412, October 1989.
[20] Hao Yuanling, and Ma Shiwei, "Fast optimization algorithm for scalar multiplication in the elliptic curve cryptography over prime field," *Lecture Notes in Computer Science*, vol. 5226, pp. 904–911, February 2008.
[21] Tao Ran, and Chen Liyan, "Fast algorithm for scalar multiplication in elliptic curve cryptography," *Transactions of Beijing Institute of Technology*, vol. 25, pp. 701–704, August 2005.
[22] Hongjun Liu, and Xingyuan Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Journal of Computational and Applied Mathematics*, vol. 59, pp. 3320–3227, October 2010.
[23] C. K. Huang, and H. H. Nien, "Multi chaotic systems based pixel shuffle for image encryption," *Optics Communications*, vol. 282, pp. 2123–2127, November 2009.
[24] T. EIGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, March 1985.
[25] N. P. Smart, "The discrete logarithm problem on elliptic curves of trace one," *Journal of Cryptology*, vol. 12, pp. 193–196, December 1999.
[26] N. K. Pareek, Vinod Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image and Vision Computing*, vol. 24, pp. 926–934, September 2006.

Xiaoqiang Zhang. He was born in 1983 in Neihuang county, Henan province, China. He is a doctor candidate at state key lab of software development environment, Beihang University. His research interests include information security, encryption theory, image encryption, digital watermarking, and software engineering.

Guiliang Zhu. He graduated at Peking University in 1977. He was born in 1950 in Zhongmou county, Henan province, China. Professor Zhu is a master supervisor at North China University of Water Conservancy and Electric Power (NCWU). His current research interests include information security, encryption algorithm, software engineering, operating system.

Weiping Wang. She is a master candidate at Department of Information Engineering, NCWU. She was born in 1985. Her research interests include software engineering, image encryption.

Mengmeng Wang. She is a master candidate at Department of Information Engineering, NCWU. She was born in 1988. Her research interests include software engineering, image encryption, encryption theory.

Shilong Ma. Professor Ma is a doctor supervisor at school of computer science and engineering, Beihang University. He was born in 1953. His research interests include calculating model on the network environment, dynamic statistic behavior of logic and computation, calculating model of massive data process, grid computing technology and application.

Flow-Based Transmission Scheduling in Constrained Delay Tolerant Networks

Zhenguo Yang[†,‡], Liusheng Huang[†,‡], Mingjun Xiao[†,‡] and Wang Liu[†,‡]
[†]Department of Computer Science & Technology, University of Science & Technology of China, China
[‡]Suzhou Institute for Advanced Study, University of Science & Technology of China, China
E-mail: {zgyang@mail, lshuang@, xiaomj@, wangliu@mail}.ustc.edu.cn

***Abstract*— Routing is one of the most challenging problems in Delay-Tolerant Networks (DTNs) due to network partitioning and intermittent connectivity. Most existing protocols are based on unlimited bandwidth and buffer size. Previous protocols based on limited bandwidth or buffer size (*e.g.*, MaxProp, RAPID, etc) only consider the scenario that each node has at most one contact opportunity during any time interval, and discuss the packet scheduling mechanisms for each separate contact to minimize the average delivery delay. However, in many applications of DTNs, more than one contact opportunities with the same source node, which are called as related contact opportunities, may arise in the same time interval. But previous scheduling algorithms cannot apply to such applications. Therefore, we propose a more general scheduling algorithm to optimize the delivery delay in constrained DTNs including the scenario that multiple related contact opportunities may arise in the same interval, called Flow-Based Transmission Scheduling (FBTS). We evaluate it on publicly available data sets against MED, MaxProp and RAPID. The results show that the delivery delay of FBTS is at least 35% shorter than that of existing works.**

***Index Terms*— DTNs, routing, flow, scheduling**

I. Introduction

Delay tolerant networks (DTNs) [1] are frequently-partitioned networks, due to high node mobility, low node density, and short radio range, etc. Opportunistic communication between nodes is the major characteristic of DTNs. Thus contemporaneous end-to-end paths may not exist in DTNs. Delay Tolerant Network Research Group (DTNRG) [2] has designed a special architecture to describe such networks, under which communication between pairwise nodes is called *contact* and packets are generally transferred in the manner of *store-carry-forward*. Because the emergence of contacts is *time-varying* and *uncertain*, researches on routing in DTNs become active and challenging.

So far, many DTN routing protocols are designed to optimize a special routing metric, such as delivery probability or delivery delay [3]–[9]. Most of these existing protocols are based on *unlimited bandwidth or buffer size*, except for MaxProp [6] and RAPID [7]. Both of them are based on limited bandwidth, which are more realistic in real DTNs. In MaxProp and RAPID, packet scheduling mechanisms are employed so that when a contact emerges, the packets with highest delivery probability or shortest delivery delay would be prior transferred, the packets with low priority would be discarded finally due to the memory constraint or the living time exhausted. However, both of them only consider the scenario that each node usually has at most one contact opportunity during any time interval such as UMassDieselNet [10] bus system, but leave another scenario out of consideration. Under the ignored scenario, multiple related contact opportunities may come up in the same time interval, which is more common especially in DTNs based on social networks [11], [12]. Figure 1 shows an example of NUS [11], where students with communication equipments (*e.g.*, motes, PDAs, etc) attend different classes in a day. The features of the scenario are as followings:

- It is a typical delay tolerant network due to the mobility of students (*e.g.*, node A might encounter with different nodes in different class sessions).
- It is common that multiple related contact opportunities may arise in the same time interval due to the gregariousness of students (*e.g.*, node B and C are both in the communication range of A in current Algorithm Class).

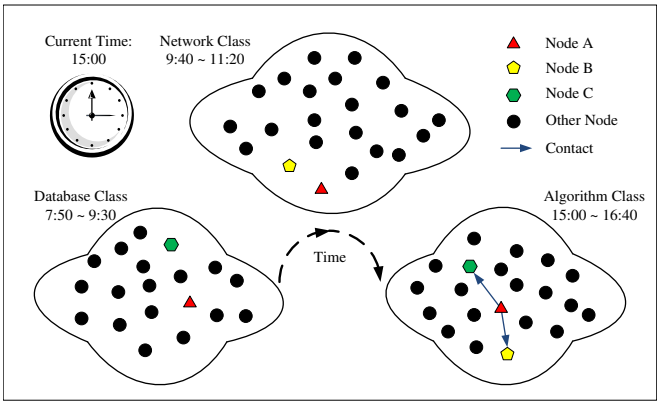


Figure 1. Students with communication equipments attend different classes in a day. They can communicate with each other when in the same class session. And current time is $15:00$, when Algorithm Class begins.

This work is supported by the National Grand Fundamental Research 973 Program of China (Grant No.2011CB302905), the National Science and Technology Major Project under Grant No. 2011ZX03005-002, the National Natural Science Foundation of China (Grant No. 60803009), and the Natural Science Foundation of Jiangsu Province in China (Grant No.BK2009150).

doi:10.4304/jcp.7.1.179-186

TABLE I.
EXPECTED DELAY (ED) AND CORRESPONDING UTILITY OF RAPID (δU) IN DIFFERENT NODES (HOUR)

	Node A		Node B		Node C	
	ED	δU	ED	δU	ED	δU
p_1	6	0	8	-2	5	1
p_2	9	0	7	2	6	3
p_3	7	0	6	1	7	0

We further present the insufficiency of previous scheduling mechanisms through above example. Students with communication equipments can form a constrained DTN, where nodes encounter with each other probabilistically (assuming the bandwidth is one packet during each class session). Node A, B and C are all in Algorithm Class at current time. Considering packet scheduling of node A, which has three packets p_1, p_2 and p_3 to deliver, our goal is to minimize the average delivery delay of these packets. We assume that the expected delay for each packet in different nodes is known in Table I. The calculation of the expected delay will be introduced in Section 2. RAPID [7] protocol schedules packets for each separate contact in descending order of δU_i, which equals the corresponding expected delay in the sender minus that in the receiver (see Table I). Figure 2(a) shows the total expected delay of RAPID is 19 hours, when scheduling packets for node B first then node C. Figure 2(b) shows another result with a smaller expected delay, which can be obtained by our transmission scheduling algorithm. From this simple example, we draw that existing protocols cannot always optimize a routing metric in constrained DTNs.

In this paper, we focus on a more general *transmission scheduling* problem in constrained DTNs. In such a transmission scheduling, packets in the sender are distributed to multiple related contact opportunities, which may arise in the same time interval, to minimize the average delivery delay. Firstly, We formalize the problem into a *linear programming* problem. Secondly, we transform the linear programming problem to a *minimum cost maximum flow (mcmf)* problem. Thirdly, we propose our transmission scheduling algorithm Flow-Based Transmission Scheduling (FBTS). Moreover, we prove the correctness of FBTS and study the problem of starvation and congestion, as well as a distributed algorithm. We also provide the improvement on them. FBTS is finally evaluated on public trace.

Since the emergence of contacts in DTNs is uncertain, transmission scheduling in DTNs is different from packet scheduling in traditional networks, where packets are scheduled to certain link. The main contributions of this paper are:

1) propose a more general transmission scheduling problem in DTNs, where multiple related contact opportunities may arise in the same time interval.
2) design a novel flow-based transmission scheduling algorithm with improvement on starvation prevention and congestion avoidance, as well as local estimation on expected delay.
3) evaluate FBTS on traces publicly available in the CRAWDAD archive [13].

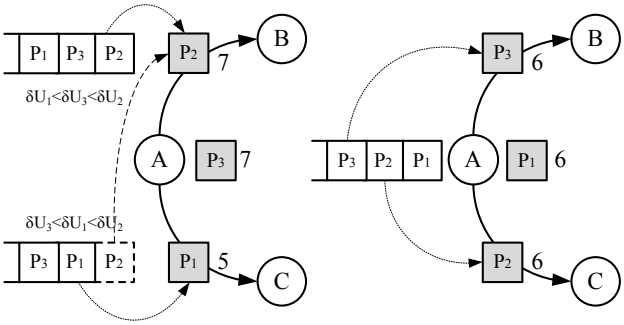


Figure 2. An example to show the insufficiency of previous scheduling

TABLE II.
SYMBOLS AND NOTATIONS OF PACKET k

Notations	Descriptions
$a(k)$	time since creation
$d(k)$	destination node
$h(k)$	current node
$q(k)$	queueing time in current node
$r(k)$	expected remaining time
$s(k)$	source node
$x(k)$	the node which has it at next time slot

The rest of this paper is organized as follows. Related Works are presented in Section 2. We provide the preliminaries including network modeling and problem formalization in Section 3. In Section 4, we propose our Flow-Based Transmission Scheduling algorithm, including network construction and transmission scheduling. The analysis and improvement are presented in Section 5, they includes correctness proof, starvation prevention, congestion avoidance and local estimation. FBTS is evaluated on NUS trace by comparing with some existing protocols in Section 5 and conclusion is presented in Section 6.

II. RELATED WORKS

In the past few years, many protocols are designed so as to minimize average delivery delay (or maximize delivery ratio). For example, Liu and Wu *et al.* proposed RCM [14] to solve the routing problem in cyclic mobispace with markov decision process and probability-based OPF [15] with optimal stopping rule. Gao *et al.* applied knapsack algorithm to solve the multicast routing problem in social network, SDM [16]. Yuan *et al.* employed a time-homogeneous semi-markov process model and proposed predict and relay (PER) [17] algorithm. Li *et al.* proposed a Social Selfishness Aware Routing (SSAR) algorithm [18] following the philosophy of design.

When considering the number of replicas, existing DTN routing protocols can be classified as either *replication-based* [3], [4], [6], [7] or *forwarding-based* [5], [9], [19]. Replication-based protocols trade huge overhead for performance, such as famous Epidemic [3] protocol, which can achieve an optimal performance at the expense of huge overhead. But huge overhead will decrease the throughput of network in constrained DTNs. Besides, most of these protocols are based on unlimited bandwidth, except for MaxProp [6] and RAPID [7]. Both MaxProp and RAPID employ packet scheduling mechanisms to schedule packets for separate contact opportunity. However, neither of them consider the scenario that multiple contact opportunities with the same source node may arise in the same time interval. Therefore, existing protocols cannot obtain an optimal metric under all scenarios of DTNs. Similar transmission scheduling problem has also been investigated in [20], which is a centralized scheduling algorithm for all packets in the network and only based on a determined model in which all contact events are pre-determined. Therefore, we need a more general scheduling algorithm.

III. Preliminaries

A. Network Model

In this paper, We focus on the effectiveness of transmission scheduling in DTNs. Therefore, we make some simple assumptions which will be addressed as part of our future work. For simplicity, time is divided into small fixed time slot, at which a contact either emerges or not. The bandwidth is limited due to short duration of contact and low bandwidth of radio. Besides, the buffer size is also limited for each node. We assume that a node can discover all other nodes within its communication range at the beginning of each time slot and two consecutive forwardings (*e.g.*,forward message x from node A to B and then to C) cannot happen in the same time slot [15]. Moreover, we assume that packets can be successfully transferred in one time slot, without regarding conflict and retransmission, etc.

Formally, we model a DTN as a widely adopted weighted network in which each pairwise nodes have different contact rates [19], referring to some real DTNs, students in NUS [11] and rollerbladers in Paris [12], for instance. The contact rate between node u and v at any time slot is denoted by $\lambda_{uv} \in [0, 1]$. The expected delay of direct delivery between node u and v is ED_{uv}. The probability of delivery which happens in this time slot is λ_{uv} and corresponding delay is 1 time slot. On the other hand, delivery happens in future with probability $1 - \lambda_{uv}$ and delay $1 + ED_{uv}$ time slots. So $ED_{uv} = \lambda_{uv} + (1 - \lambda_{uv}) \times (1 + ED_{uv})$, $ED_{uv} = 1/\lambda_{uv}$ [9], [19]. Further, expected minimum delay is the expected time an optimal opportunistic routing scheme takes from source u to destination v, denoted by $EMD(u, v)$. The calculation method of EMD is in [5], where a shortest path algorithm is adopted. The transmission bandwidth between node u and v is defined as the number of packets that can be transferred via corresponding contact during one time slot, denoted by $B(u, v)$. Given a node u, we use $C(u)$ to denote its current available buffer size. For a given packet k, there are some definitions in Table II.

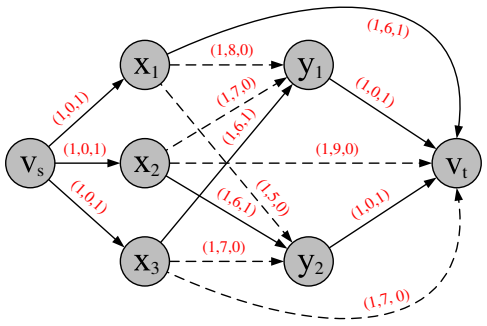


Figure 3. Cost Flow Network and corresponding Minimum Cost Maximum Flow of node A in Algorithm Class shown in Figure 1. Vertex x_i denotes the i^{th} packet in A and node y_j denotes the j^{th} neighbor of A. The red expression on each edge (c, w, f) denotes the capacity, cost, and flow of minimum cost maximum flow respectively.

B. Problem Formalization

Based on previous works, we can identify several desirable design goals for a transmission scheduling scheme in constrained DTNs. Specifically, an efficient transmission scheduling algorithm proposed in this context should distribute packets to multiple related contact opportunities to minimize the average delivery delay. In this section we formalize the problem into a *linear programming problem*.

Before the formalization, we define some notations to describe the network parameters. Considering node u at current time slot, $P_u = \{p_u^i | 1 \leq i \leq n_u\}$ denotes the packets to forward. The nodes which are within the communication range of node u are called its neighbors, denoted by $C_u = \{c_u^j | 1 \leq j \leq m_u\}$. Because packets and neighbors are different at each time slot, we only define them at current time slot. In addition, we use $forward_{i,j} \in \{0, 1\}$ to indicate whether node u forwards packet p_u^i to node c_u^j in current time slot. And $forward_{i,j} = 1$ means forwarding, otherwise not. So

$$forward_{i,j} = 0 \text{ or } 1, \quad 1 \leq i \leq n_u, 1 \leq j \leq m_u \quad (1)$$

Each packet has only one copy in the network during the transmission from the source to the destination. That is

$$0 \leq \sum_{j=1}^{m_u} forward_{i,j} \leq 1, \quad 1 \leq i \leq n_u \quad (2)$$

Besides, due to the constraint of bandwidth, the number of packets that a neighbor might receive during current time slot is not more than the bandwidth between them. We have

$$0 \leq \sum_{i=1}^{n_u} forward_{i,j} \leq B(u, c_u^j), \quad 1 \leq j \leq m_u \quad (3)$$

The expected delay of a packet is composed of two parts: the time since creation and the expected remaining time. Then

$$D(p_u^i) = a(p_u^i) + r(p_u^i), \quad 1 \le i \le n_u \tag{4}$$

We use the expected minimum delay of a packet between the node which will have the copy at next time slot and its destination to estimate the expected remaining time of it. So

$$r(p_u^i) = EMD(x(p_u^i), d(p_u^i)), \quad 1 \le i \le n_u \tag{5}$$

The main purpose of our transmission scheduling is to minimize the average expected delay, which is equivalent to the minimum of total expected delay of all packets. That is

$$\min \sum_{i=1}^{n_u} D(p_u^i) \tag{6}$$

Up to now, we have formalized transmission scheduling problem in constrained DTNs into a linear programming problem. In next section, we will solve it further by transforming it to a minimum cost maximum flow problem.

IV. Transmission Scheduling Algorithm

To solve the linear programming problem above, we transform it to corresponding *minimum cost maximum flow* problem first. Then, we propose our transmission scheduling algorithm based on *mcmf*.

A. Cost Flow Network Construction

Before the construction, we present some definitions of cost flow network, denoted by $G = (V, E)$. Each edge (a, b) has capacity $c(a, b)$, flow $f(a, b)$ and cost $w(a, b)$. We use $\mathcal{A}$ to denote the minimum cost maximum flow of G, where $f(\mathcal{A})$ and $w(\mathcal{A})$ are equal to the maximum flow and the minimum cost respectively.

Accordingly, every node in DTN has a corresponding cost flow network at each time slot, where flow is equivalent to the distribution strategy and cost is equivalent to corresponding expected delay. Considering node u at current time slot, the construction of $G = (V, E)$ is as followings:

$\underline{V}$ is consisted of a source v_s, a sink v_t and two independent vertex sets $X = \{x_i | 1 \le i \le n_u\}$ and $Y = \{y_j | 1 \le j \le m_u\}$, where x_i denotes packet p_u^i and y_j denotes node c_u^j respectively.

$\underline{E}$ is consisted of four categories of edges as follows:

- (v_s, x_i), an edge from source v_s to each vertex $x_i \in X$, with one capacity and zero cost. So

$$c(v_s, x_i) = 1, \quad w(v_s, x_i) = 0 \tag{7}$$

- (x_i, v_t), an edge from each vertex $x_i \in X$ to sink v_t with one capacity, whose cost equals to the expected remaining time of packet p_u^i in node u. So

$$c(x_i, v_t) = 1, \quad w(x_i, v_t) = EMD(u, d(p_u^i)) \tag{8}$$

- (x_i, y_j), an edge from vertex x_i to vertex y_j with one capacity, whose cost equals to the expected remaining time of packet p_u^i in node c_u^j. So

$$c(x_i, y_j) = 1, \quad w(x_i, y_j) = EMD(c_u^j, d(p_u^i)) \tag{9}$$

- (y_j, v_t), an edge from each vertex $y_j \in Y$ to sink v_t with zero cost, whose capacity is equal to the minimum value between transmission bandwidth from node u to node c_u^j and the available buffer size for node c_u^j. So

$$c(y_j, v_t) = \min\{B(u, c_u^j), C(c_u^j)\}, \quad w(y_j, v_t) = 0 \tag{10}$$

So far, a flow network $G = (V, E)$ for node u at current time slot has been constructed. Figure 3 shows the construction of node A introduced in Section 1.

B. Flow-Based Transmission Scheduling

Minimum cost maximum flow problem is a very classic problem. The surveys by Ahuja, Magnanti, and Orlin [1989,1991] and by Goldberg, Tardos, and Tarjan [1989] provide details concerning this field. Because some existing algorithms are based on nonnegative integer, we can transform the rational number of cost to nonnegative integer by adding a positive number first then multiplying by a suitably large number, which does not affect our results. The computational complexity of minimum cost maximum flow procedure is $O\big((n+m+2)^3\big)$, where n is the number of packets and m is the number of related contacts. The corresponding minimum cost maximum flow $\mathcal{A}$ is shown in Figure 3 by solid lines.

After the flow procedure, we distribute packets according to the flow result $\mathcal{A}$. We will distribute packet p_u^i to node c_u^j if $f(x_i, y_j) = 1$. Packet p_u^i will not be transferred on condition that $f(x_i, v_t) = 1$. The integrated algorithm is formally described in Algorithm 1.

V. Analysis and Improvement

To analyze the effectiveness of FBTS, we firstly prove the correctness of it. Then, we discuss the problem of starvation and congestion and provide the improvement on starvation prevention and congestion avoidance.

A. Correctness Proof

According to the relationship between linear programming and minimum cost maximum flow, the cost flow network we constructed satisfies the linear constraints in Section 3. Therefore, our work is to prove the optimality of FBTS based on corresponding minimum cost maximum flow.

Theorem 1: The average expected delay of FBTS is minimal according to corresponding *minimum cost maximum flow*.

Proof: considering node u at current time slot, obviously, the maximum flow of corresponding $\mathcal{A}$ is $f(\mathcal{A}) = n_u$. Thus, there is one flow out of each vertex $x_i \in X$. $f(x_i, y_j) = 1$ means node u will forward packet

Algorithm 1 Flow-Based Transmission Scheduling Algorithm

Require: $EMD, B, u, t.$
Ensure: $forward.$
for $i = 1$ to n_u **do**
  $c(v_s, x_i) = 1, w(v_s, x_i) = 0.$
  $c(x_i, v_t) = 1, w(x_i, v_t) = EMD(u, d(p_u^i)).$
end for
for $j = 1$ to m_u **do**
  $c(y_j, v_t) = B(u, c_u^j), w(y_j, v_t) = 0.$
end for
for $i = 1$ to n_u **do**
  for $j = 1$ to m_u **do**
    $c(x_i, y_j) = 1, w(x_i, y_j) = EMD(c_u^j, d(p_u^i)).$
  end for
end for
Adopt minimum cost maximum flow algorithm
for $i = 1$ to n_u **do**
  for $j = 1$ to m_u **do**
    $forward_{i,j} = f(x_i, y_j).$
  end for
end for

p_u^i to node c_u^j. And $f(x_i, v_t) = 1$ means packet p_u^i will not be transferred in this time slot. Since the cost of $\mathcal{A}$ is:

$$w(\mathcal{A}) = \sum_{f(x_i,y_j)=1} w(x_i, y_j) + \sum_{f(x_i,v_t)=1} w(x_i, v_t) \quad (11)$$

According to (8) and (9), then

$$w(\mathcal{A}) = \sum_{i=1}^{n_u} EMD(x(p_u^i), d(p_u^i)) \quad (12)$$

Because $w(\mathcal{A})$ is minimal and $\sum_i a(p_u^i)$ is only related to the given time slot, $\sum_i D(p_u^i) = \sum_i a(p_u^i) + w(\mathcal{A})$ is minimal. Therefore, we draw the conclusion that the average expected delay is minimal in FBTS according to corresponding minimum cost maximum flow. ■

B. Starvation Prevention

Due to the low priority, some packets might not be scheduled to relay nodes till expiring of TTL, on condition that packets with higher priority always exist in queue. Therefore we use a special utility to prevent starvation.

Definition 1: (Transfer Priority) Transfer priority is a per-packet priority utility derived for preventing starvation of packets, which increases as the queuing time increases with factor α, denoted by $TP(k)$ for packet k.

$$TP(k) = \alpha \cdot q(k), \quad \alpha \in (0, 1]$$

Considering node u, the method of preventing starvation is to modify the cost of edges out of vertex $x_i \in X$ by subtracting corresponding $TP(p_u^i)$. Because *Transfer Priority* is only related to the queuing time, we can easily draw that the modification does not affect the correctness of FBTS. So new cost of $\mathcal{F}$ becomes:

$$w(\mathcal{F}) = \sum_{i=1}^{n_u} EMD(x(p_u^i), d(p_u^i)) - \alpha \times \sum q(p_i)$$

$\sum q(p_i)$ is certain at time t, so the modification does not affect the correctness of our scheme. Besides, we can get another two corollaries:

- The possibility of distributing a packet to relay nodes increases as its queuing time increases.
- The packets with same destination are transferred in queuing order.

As queuing time of packet p_u^i increases, corresponding $w(x_i, y_j)$ decreases. Accordingly, the possibility of augmenting a flow through edge (x_i, y_j) increases. The second corollary can be proved by the reverse proving. If two packets with the same destination are not transferred in queuing order, we could exchange their distribution strategies to obtain a better delivery delay. Through our simulation, we find that $\alpha = 0.1$ can achieve better results.

C. Congestion Avoidance

The queuing time of a packet will be long when many packets with the same destination are in front of it. Therefore, we introduce the improvement on avoiding congestion if many same destination packets are in queue.

Definition 2: (Expected Transfer Speed) For the packets with the same destination v in node u, Expected Transfer Speed is the expected number of them that can be transferred during one time slot, denoted by $ETS(u, v)$. Let $RS_{uv} = \{z | EMD(z, v) < EMD(u, v)\}$ denote the nodes whose expected delay is smaller than that in u, so

$$ETS(u, v) = \sum_{z \in RS_{uv}} \lambda_{u,z} \cdot B(u, z)$$

Definition 3: (Expected Queueing Time) Expected Queueing Time is the expected time a packet will spend on queuing in current node, denoted by $EQT(k)$ for packet k. Let N_k denote the number of the same destination packets queuing in front of k, then

$$EQT(k) = N_k / ETS(h(k), d(k))$$

Considering node u, the method of avoiding congestion is to modify the cost of each edge (x_i, v_t) by adding corresponding $EQT(p_u^i)$. So the possibility of distributing the packets queuing in the back to other emerged nodes increases (if there is redundant bandwidth).

D. Local Estimation

In this sub-section, we investigate an estimation of EMD based on the partial information which can be obtained directly by a given node. We get help from "ego network", which is used as local data structure to maintain the partial information in social network analysis. Ego networks [21] consist of a focal node ("ego") and the nodes to whom ego is directly connected to (these are called "alters") plus the ties, if any, among the alters. Therefore, each node is responding for maintaining the average delay of inter-meeting for every neighbor (D_i for the i^{th} neighbor). Besides, we also define an associated value of delay for each node (DV), which is used to predicate the delay of future meeting. To track the delay of inter-meeting, a node also needs to maintain its latest

meeting time for each neighbor, denoted by T_i for the i^{th} neighbor. Hence, the value of average delay and meeting time are updated when the node encounters with a neighbor, and its corresponding ego network also needs to be updated. When two nodes encounter with each other, they firstly update their local data, and then exchange their latest local data, finally update their local data again.

Given node v, for each message $m \in P_v$ in the queue, if the destination of m is one neighbor of v, namely $d(m) \in S_v$, for every neighbor $u \in S_v$, we will use the delay of inter-meeting between u and the destination $d(m)$ to approximate the expected remaining delay of message m on node u. Otherwise, we assume each neighbor can deliver m to its destination with different delivery delay, which is related to respective delay value DV, with a factor D_{init}. Therefore, given a node v

$$\tilde{EMD}(u, d(m)) = \begin{cases} EMD(u, d(m)) & \text{if } d(m) \in N_v \\ DV_u \cdot D_{init} & \text{otherwise} \end{cases}$$

Besides, different D_{init} can lead to different results. According to our simulation on different D_{init}s based on *NUS Trace*, D_{init} equals to 20 can achieve a better delivery, seen figure 4, on condition that bandwidth equals to 40 pkts/slot and buffer size is infinite.

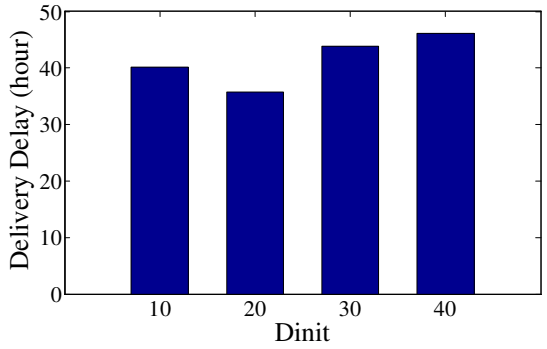


Figure 4. Delivery Delay versus D_{init} in NUS

VI. SIMULATION

In this section, we evaluate our FBTS with above improvement against three other protocols by using a wide variety traces: student contact patterns in NUS [11]. The simulator we experiment on is developed ourselves in JAVA on Eclipse, which is similar to a prevailing DTN simulator, ONE [22].

A. Protocols in comparison

The protocols in comparison are MED [5], MaxProp [6] and RAPID [7]. For fairness in the comparison, we use their enhanced versions which make use of the same level of prior knowledge of historical connectivity patterns as FBTS does. In our copy-controlled enhanced versions, all of them are based on *single-copy*. We use FIFO mechanism for MED. Similar experimental design can be seen in [14], [15].

TABLE III.
SETTINGS FOR NUS TRACE

parameter name	default or range
number of sessions	30
number of students	100
length of time slot	1 hour
time-to-live of packet	40 hours
simulation time	400 hours
packet production rate	5∼40 packets per time-slot
transmission speed (bandwidth)	10∼80 packets per time-slot

MED [5]. It is a source routing protocol based on the same level of prior knowledge as FBTS. The difference is MED fails to exploit superior edges which become available after the route has been computed.

MaxProp [6]. This protocol is based on limited bandwidth. MaxProp takes f_j^i to denote the probability of the next meeting node of node i is j. The delivery probability from a source to a destination is the total cost on their shortest path. In the enhanced version MaxProp*, we let $f_j^i/f_k^i = \lambda_{ij}/\lambda_{ik}$.

RAPID [7]. This protocol is also based on limited bandwidth. RAPID forwards packets with the highest value of δU_k, which denotes the increase in U_k by forwarding packet k. The corresponding utility U_k is the negative value of expected delay. In the enhanced version RAPID*, U_k equals the time since creation plus the remaining expected delay.

B. Simulation on NUS

Accurate information of human contact patterns is available in several scenarios such as university campuses. Student contact patterns in National University of Singapore (NUS) [11] were inferred from the information on class schedules and class rosters for the Sprint semester of 2006 in which there were 22341 students and 4885 class sessions included. If one knows class schedules and student enrollment for each class on a campus, accurate information about contact patterns between students over large time scales can be obtained without a long-term contact data collection. The advantage of the trace is that it exhibits a DTN based on a real social network and provides contact patterns of a large population over a long period. Many existing protocols [14], [15] are evaluated on this trace.

We select several class sessions M and a number of students N in each experiment. Contacts related to non-selected students or non-selected class sessions are ignored. Assume each student attends a class with a attendance probability 0.8 and sends packets to others randomly at each time slot. We define the number of packets generated by each student every time slot as *production rate*.

Our data processing includes the following steps: (1) The selection of M class sessions. If they are selected randomly, we cannot guarantee the connectivity of network. Thus we design a new selection method as follows. The first session is selected randomly. We select the k^{th} session c_k as the one with the highest score

$\sum comm(c_i, c_k)$ among the sessions that are not yet selected and not conflict with selected sessions in fields of time, where the function $comm$ is defined as the number of common students enrolled in both class sessions. (2) The selection of N students is in decrease order of the number of selected class sessions each student enrolled in. To guarantee the connectivity of network, the candidate must have some contacts with selected students (except the first student). (3) Calculate encounter probability of each pair of selected students at each class session.

C. Results of Simulation

In the simulation, we evaluate Delivery Ratio, Delay and Hop versus bandwidth (from 10 to 80 packets per time-slot), packet production rate (from 5 to 40 packets per hour) and buffer size (from 1000 to 8000 packets per node). The detailed simulation settings for NUS trace are shown as Table III. Figure 5 illustrates delivery ratio, delay and hop versus bandwidth, with default TTL (48 time-slots) and packet production rate (10 pkts/hour). The delivery ratio is approximately 24% greater than RAPID and 55% greater than MaxProp. For the delivery delay, the other three protocols are at least 56% longer than FBTS. The hop of FBTS is the smallest in four, whose mean value is 2.4. Figure 6 presents these three metrics versus packet production rate on condition that the bandwidth is 40 pkts/time-slot and the buffer size is infinite. Compared with three other protocols, FBTS shorten at least 30% on delivery delay. When the TTL equals to 48 hours, FBTS can achieve 70.7%, 57.5% and 31.2% more than MED, MaxProp and RAPID, respectively. The average hop of FBTS is 2.4, which is the smallest in all. These metrics versus buffer size are shown on Figure 7. The delivery delay of MED is 53.1% longer than FBTS on average, and MaxProp and RAPID obtain 65.0% and 52.0% longer, respectively. If TTL is 48 hours, the delivery ratio of FBTS is 39.8%, 47.7% and 21.2% greater than MED, MaxProp and RAPID, respectively. At the mean time, the smallest hop is also obtained by FBTS, whose value is 2.8 on average.

VII. Conclusion

In this paper, we first investigated a more general transmission scheduling problem, and then formalized the problem into a *linear programming* problem, which was transformed to a *minimum cost maximum flow* problem further. Finally, we proposed our *Flow-Based Transmission Scheduling* algorithm to minimize the average delivery delay, which can apply to more applications of DTNs, including the scenario that multiple related contact opportunities may arise in the same time interval. In addition, we proved the correctness of FBTS and provided the improvement on starvation prevention and congestion avoidance, as well as estimation on the expected delivery delay locally. Moreover, FBTS was evaluated on the real DTNs traces against MED, MaxProp and RAPID. And the simulation results verified the efficiency of FBTS in delivery ratio, delay and hop.

Acknowledgment

This work is supported by the National Grand Fundamental Research 973 Program of China (Grant No.2011CB302905), the National Science and Technology Major Project under Grant No. 2011ZX03005-002, the National Natural Science Foundation of China (Grant No. 60803009), and the Natural Science Foundation of Jiangsu Province in China (Grant No.BK2009150).

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, 2003.
[2] "Delay Tolerant Network Research Group," http://www.dtnrg.org/.
[3] A. Vahdate and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep. CS-2000-06, June 2000.
[4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings WDTN*, 2005.
[5] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *Proceedings WDTN*, 2005.
[6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *IEEE INFOCOM*, 2006.
[7] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *ACM SIGCOMM*, 2007.
[8] S. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *IEEE INFOCOM*, 2009.
[9] M. Xiao, L. Huang, Q. Dong, A. Liu, and Z. Yang, "Optimal opportunistic routing in probabilistically contacted delay tolerant networks," *Journal of Computer Science and Technology*, vol. 24, no. 5, pp. 975–986, 2009.
[10] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing," in *ACM MobiCom*, 2007.
[11] V. Srinivasan, M. Motani, and W. T. Ooi, "Analysis and Implications of Student Contact Patterns Derived from Campus Schedules," in *ACM MobiCom*, 2006.
[12] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *IEEE INFOCOM*, 2009.
[13] "CRAWDAD data set," http://crawdad.cs.dartmouth.edu/.
[14] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *ACM MobiHoc*, 2007.
[15] ——, "An optimal probabilistic forwarding protocol in delay tolerant networks," in *ACM MobiHoc*, 2009.
[16] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *ACM MobiHoc*, 2009.
[17] I. C. Q. Yuan and J. Wu, "Predict and relay: an efficient routing in disruption-tolerant networks," in *ACM MobiHoc*, 2009.
[18] S. Z. Q. Li and G. Cao, "Routing in socially selfish delay tolerant networks," in *IEEE INFOCOM*, 2010.
[19] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 773–782, 2008.
[20] D. Hay and P. Giaccone, "Optimal routing and scheduling for deterministic delay tolerant networks," in *Wireless On-Demand Network Systems and Services*, 2009.

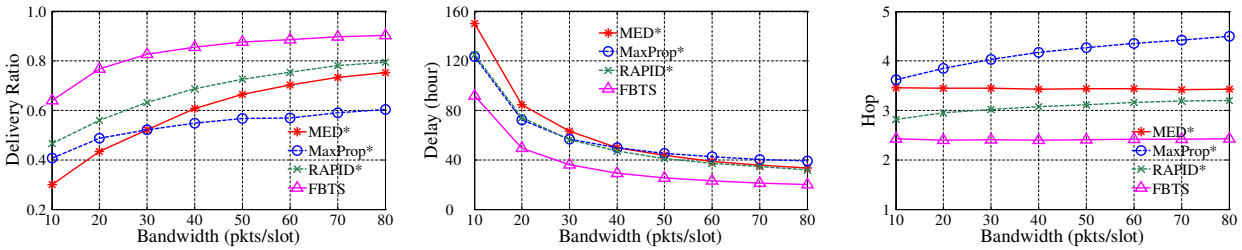


Figure 5. Delivery Ratio, Delay and Hop versus bandwidth (packets per time-slot) in NUS

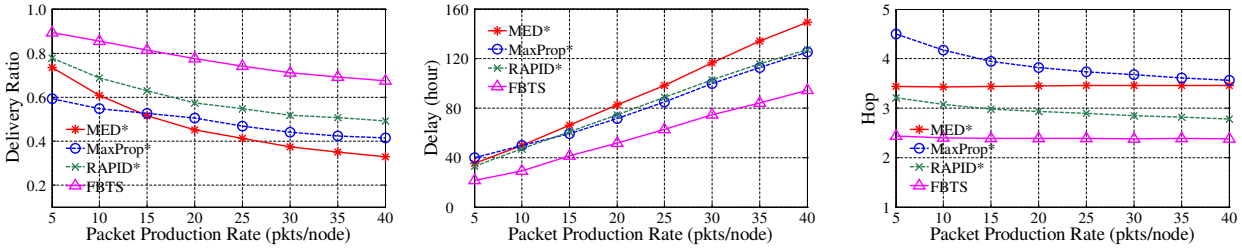


Figure 6. Delivery Ratio, Delay and Hop versus packet production rate (packets per time-slot) in NUS

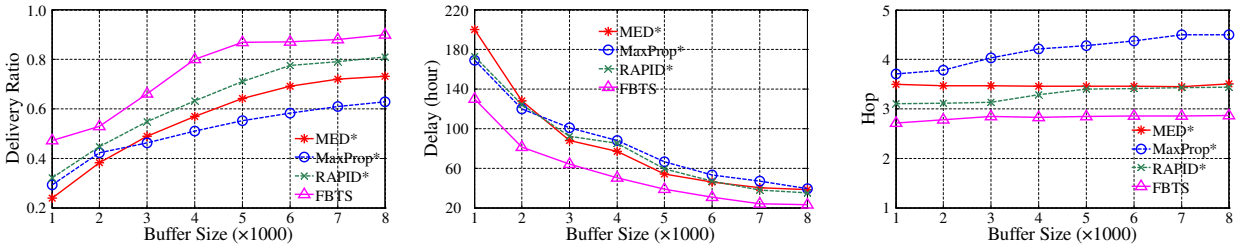


Figure 7. Delivery Ratio, Delay and Hop versus buffer size (packets per node) in NUS

[21] E. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 5, pp. 606 –621, may 2009.

[22] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. New York, NY, USA: ICST, 2009.

Zhenguo Yang, born in 1984, is currently a Ph.D. candidate in the Department of Computer Science and Technology, University of Science and Technology of China. His main research interests include wireless sensor networks and delay tolerant networks. E-mail:zgyang@mail.ustc.edu.cn.

Liusheng Huang was born in Anhui Province, China, in 1957. He received the M.S. degree in computer science from University of Science and Technology of China in 1988. He is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at the University of Science and Technology of China. He has published 6 books and more than 90 papers. His research interests are in the areas of wireless sensor networks, information security, distributed computing and high performance algorithms.

Mingjun Xiao , born in 1976. Received his Ph.D. degree from the Department of Computer Science and Technology, University of Science and Technology of China in 2004. Lecturer, member of China Computer Federation. His major research interests include wireless sensor network and information security.

Wang Liu, born in 1985, is currently a Ph.D. candidate in the Department of Computer Science and Technology, University of Science and Technology of China. His main research interests include throughput capacity, mobility, delay within Wireless Mesh Network and Wireless Ad-Hoc Network.

Tree Based Orthogonal Least Squares Regression with Repeated Weighted Boosting Search

Lihua, Fu, Hongwei Li*
School of Mathematics and Physics, China University of Geosciences, Wuhan 430074, China
Email:fulihua9270@yahoo.com.cn, hwli@cug.edu.cn

Meng Zhang
Department of Computer Science, Central China Normal University, Wuhan 430079, China
Email: morosezhang@yahoo.com.cn

***Abstract*—Orthogonal Least Squares Regression (OLSR) selects each regressor by repeated weighted boosting search (RWBS). This kind of OLSR is known to be capable of producing a much sparser model than many other kernel methods. With the aid of tree structure search, this paper is to construct an even sparser regression model in the framework of OLSR with RWBS. When RWBS being used to solve the optimization at each regression stage, OLSR is extended by keeping the k (k >1)excellent regressors, which minimize the modeling MSE, rather than only choose the best one at each iteration. In this way, the next regressor will be searched in k subspaces instead of in only one subspace as the conventional method. Furthermore we propose a subtree search to decrease experimental time complexity, by specifying the total number of children in every tree depth. The new schemes are shown to outperform the traditional method in the applications, such as component detection, sparse representation for ECG signal and 2-d time series modeling. Besides, experimental results also indicate that subtree based algorithm is with much lower time complexity than tree based one.**

***Index Terms*—Orthogonal least squares, repeated weighted boosting search, tree structure search**

I. INTRODUCTION

A basic principle in nonlinear data modeling is the parsimonious principle of ensuring the smallest possible model that explains the training data. The state-of-art sparse kernel modeling techniques, such as support vector machines (SVM) and relevance vector machine (RVM) [1,2] have widely been adopted in data modeling applications.

However, most existing kernel methods use the fix parameter for every regressor. For example, standard SVM with Gaussian kernel generally adopts a fix scale parameter for each term by cross-validation. For the non-flat functions which contain both the steep variations and the smooth variations, the fix scale scheme will damage the sparsity of SVM.

Ref. [3] takes this problem as a linear regression in a combined feature space which is implicitly defined by a set of translation invariant kernels with different scales. It was reported that the multiscale SVM could produce much sparser kernel model than the conventional methods. However, multiscale SVM needs to construct a candidate set for scales, which is not hard task for users.

The orthogonal least squares regression(OLSR) algorithm [4] developed in the late 80s for nonlinear system modeling, remains popular for nonlinear data modeling, for the reason that the algorithm is simple and efficient, and is capable of producing very sparse regression model with good generalization performance. Over the time, many improved variants of the OLSR algorithm have been proposed [5-9]. OLSR can tune the scales in every term.

In order to minimize the modeling MSE, at each stage, OLSR selects the best regressor in a subspace which is orthogonal to the linear space spanned by the already selected regressors. Different regressor will lead to different subspace for the next regressor being searched. Thus the selection of every regressor will dramatically affect the choosing of regressor in the next stage. Nevertheless OLSR is a greedy algorithm, which only seeks the best performance in the current stage, and ignores the affect on the next stage. To get a good performance, OLSR needs more regressors than necessary.

This paper extends OLSR by keeping the k excellent regressors which minimize the modeling MSE, rather than ignoring them after getting the best one at each regressor stage. These surviving regressors (regarded as nodes) are used to calculate the k new subspaces and corresponding residuals. Because each node creates k children, this tree-structure search scheme, in the L-th level, involves k^L best nodes rather than only one node in the conventional method (see Fig. 2(a)). In order to decrease experimental time complexity, we specify the total number of children as s in every level (see Fig. 2(b)). For notation clarity, this paper terms the former structure as tree (k) and the latter as subtree(s, k).

In fact, one can implement the tree structure search in many variants of the OLSR algorithm. This paper only considers Ref. [9]. In [9], the optimization at each regression stage is carried out with a simple search algorithm re-enforced by boosting, termed as repeated weighted boosting search (RWBS). In this way, OLSR tunes the centre vector and diagonal covariance matrix of individual regressor by incrementally minimizing the training mean square error (MSE). This kind of OLSR can produces a much sparser regression model than many

*corresponding author

doi:10.4304/jcp.7.1.187-195

other kernel machines and previous versions of OLSR [9-11], largely because of its flexibility to select its optimal scales.

RWBS is a guided global search algorithm which involves the excellent individual, obtained at the previous iteration, into the new iteration. This scheme makes the algorithm constringency. At each regressor stage, OLSR with RWBS always keeps the optimal solution as the newly selected regressor parameters, and the other excellent individuals mentioned above are ignored.

With the aid of tree(k) and subtree(s, k), an extension for OLSR with RWBS is presented. Because the tree structure search concerns with the performance not only in the current stage, but also in a more global view than conventional scheme, this method will lead to a meaningful improvement to the conventional OLSR.

Experimental results show that both tree based and subtree based OLSR create a much sparser signal representation than conventional OLSR. And subtree based algorithm can avoid the exponentially increasing computational load caused by tree based OLSR.

Tree based algorithm for regression also can be found in matching pursuit field, where a sparse linear combination is searched in a given linear space with finite dimension [12-13].

II. OLSR WITH RWBS

A. Theory

Given N pairs of training data $\{\mathbf{x}(l), y(l)\}_{l=1}^{N}$ and kernel function $\varphi(\cdot,\cdot)$, let

$$\mathbf{\Phi}_i = [\varphi(\mathbf{u}(i), \mathbf{x}(1)), \cdots, \varphi(\mathbf{u}(i), \mathbf{x}(N))]^T, \ i = 1, 2, \cdots, M \quad (1)$$

where M denotes the size of model, $\mathbf{u}(i)$ is the parameter vector, thus the regression matrix is $\mathbf{\Phi} = [\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_M]$, weight vector $\mathbf{w} = [w_1, \cdots, w_M]^T$, output vector $\mathbf{y} = [y(1), \cdots, y(N)]^T$, and error vector $\mathbf{e} = [e(1), \cdots, e(N)]^T$. The regression model can be presented as following matrix form

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \mathbf{e} \quad (2)$$

The goal of modeling data is to find the best linear combination of the column of $\mathbf{\Phi}$ (i.e. the best value for $\mathbf{w}$ and $\mathbf{u}(i)$) to explain $\mathbf{y}$ according to some criteria.

By OLSR algorithm, the solution is searched in a transformed orthogonal space. In more detail, let an orthogonal decomposition of the regression matrix $\mathbf{\Phi}$ be $\mathbf{\Phi} = \mathbf{HA}$, where $\mathbf{A}$ is an upper triangular matrix with the unit diagonal element and $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_M]$ with the orthogonal columns that satisfy $\mathbf{H}_i^T \mathbf{H}_j = 0$ if $i \neq j$. The regression model (3) can alternatively be expressed as

$$\mathbf{y} = \mathbf{H}\boldsymbol{\theta} + \mathbf{e} \quad (3)$$

where the new weight vector $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_M]^T$ satisfies the triangular system $\boldsymbol{\theta} = \mathbf{Aw}$. For the orthogonal regression model, the training MSE can be expressed as

$$J = \mathbf{e}^T\mathbf{e}/N = \mathbf{y}^T\mathbf{y}/N - \sum_{i=1}^{M} \mathbf{H}_i^T\mathbf{H}_i\theta_i^2/N \quad (4)$$

Thus the training MSE for the L -term subset model can be expressed as $J_L = J_{L-1} - \mathbf{H}_L^T\mathbf{H}_L\theta_L^2/N$ with $J_0 = \mathbf{y}^T\mathbf{y}/N$.

At the L-th stage of regression, the L-th regressor is determined by maximizing the error reduction criterion $E_L = \mathbf{H}_L^T\mathbf{H}_L\theta_L^2/N$ with respect to the kernel parameter $\mathbf{u}(L)$. Generally, Gaussian kernel is often the first choice of kernel because of its excellent generalized ability.

B. Algorithm

Some guided random search methods can be used to determine the parameters of the k-th kernel regressor, such as the genetic algorithm and adaptive simulated annealing. RWBS is recently proposed global searching algorithm. It is extremely simple and easy to implement, involving a minimum programming effort. Before implementing RWBS, several parameters need be set, such as the initial size of population Ps, the generation of outer loop NG, and the iteration of inner loop Nb. One can refer to [9,14] for more detail.

When searching the L-th regressor of OLSR with RWBS, the algorithm can be described as Fig. 1

NOTE 1

RWBS can get the global optimization by involving the good individual $\mathbf{u}_t^L, t = 1, \cdots, NG-1$, obtained at the previous iteration, into the new iteration. Always the global optimization $\mathbf{u}_{NG}^L$ is kept as the newly selected regressor parameter, and all other $\mathbf{u}_t^L, t = 1, \cdots, NG-1$ are ignored.

NOTE 2

There are other decision criterion to stop the iteration besides t=NG. But for simplicity, we let the condition t=NG be the sole criterion.

III. TREE BASED ALGORITHM

A. Theory

In this section, the tree based OLSR is described. The algorithm also has a recursive structure. At L-th regressor stage, when implementing RWBS in Fig.1, we select k best vectors from all of the good individuals $\mathbf{u}_t^L, t = 1, \cdots, NG$, to minimize the training MSE. That is

$$\tilde{\mathbf{u}}_i^L = \arg\min_{\mathbf{u}_t^L} J(\mathbf{u}_t^L), \quad \text{and } \tilde{\mathbf{u}}_i^L \neq \{\tilde{\mathbf{u}}_1^L, \tilde{\mathbf{u}}_2^L, \cdots, \tilde{\mathbf{u}}_{i-1}^L\}, \quad i = 1, \cdots, k \quad (5)$$

where $k < NG$. Thus each node can create k child nodes $\tilde{\mathbf{u}}_i^L, i = 1, \cdots, k$. The selected vectors are treated as the candidates for the parameter vector for the L-th regressor. For each $\tilde{\mathbf{u}}_i^L$, corresponding regressor $\mathbf{\Phi}_i^L$ and training error J_i^L are also calculated according to (1) - (4). Once the required tree's depth is reached or the training MSE is small enough, the program is broken and we keep the combination of nodes which produces the best performance as the final solution. We term the algorithm as tree(k) which lets each node creates k new individuals.

As Fig 2(a) shows, there are k^L branches when the tree depth reaches L, which means that we will search the solution in the k^L subspaces. Even if one let k be a

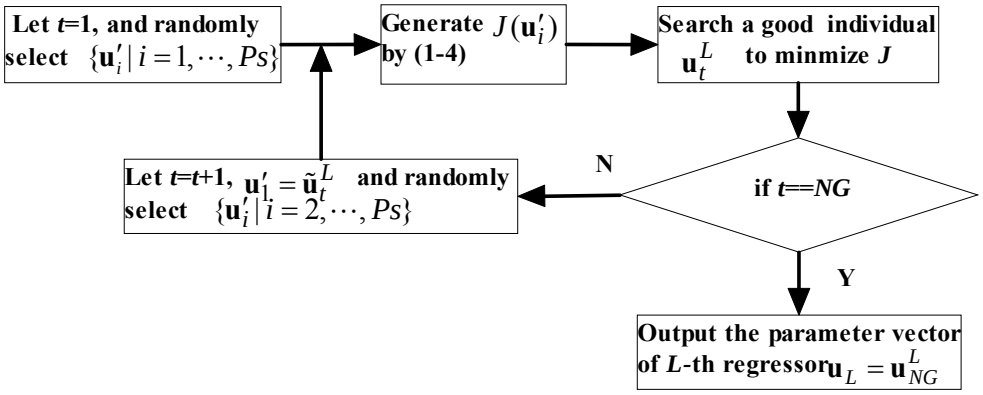


Figure 1. The scheme of OLSR with RWBS

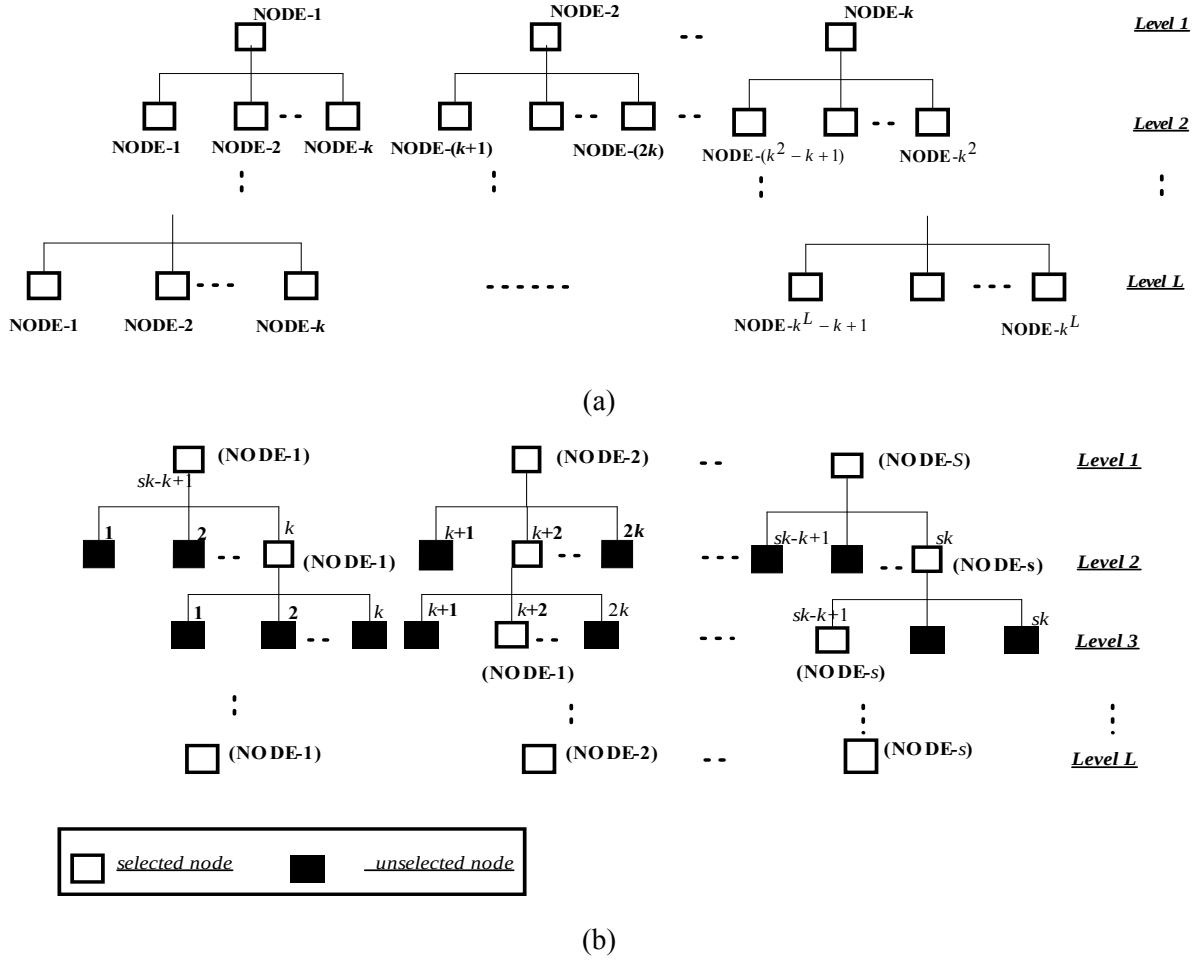


Figure 2. (a): tree(*k*) structure: every node can produce *k* child nodes. (b):sub-tree(*s*, *k*) structure, in each level, *s* nodes are selected to produce new nodes. Each selected node can produce *k* child nodes.

moderate number, the load of computation is prohibitive. This paper also proposes subtree(*s*, *k*) algorithm, which also lets each node have *k* children, but only s best individuals is kept in each level (see Fig. 2(b)), Tree(s, k) avoids the exponential increase of nodes.

Given the thresholds for tree depth *N* and training accuracy ξ, and let every node can generate *k* children. We depict the tree based and subtree based algorithm as following:

For $b = 1 : r(L-1)$ (the positive integer $r(L)$ denotes the total number of new surviving individual generated at

level L , and initial value $r(0)=k$ for tree(k) or $r(0)=s$ for subtree(s, k)).

With the aid of RWBS, generate the candidate parameter column vectors $\tilde{\mathbf{u}}_i^L, i=1,\cdots,k$ for L-th regressor according to (5). And then create the candidate parameter matrix for the regression model with L terms, $\tilde{\mathbf{u}}\_candidate_{b,i}^L=[\mathbf{u}\_candidate_b^{L-1}, \mathbf{u}_i^L], i=1, 2,\cdots,k$ where $\mathbf{u}\_candidate_b^{L-1}$ denotes the candidate parameter matrix for the model with $L-1$ terms. And the initial element $\mathbf{u}\_candidate_b^0$ is a null matrix. Calculate $J_{b,i}^L=J(\tilde{\mathbf{u}}\_candidate_{b,i}^L)$ according to the equations (1-4).

End for

$\tilde{\mathbf{u}}\_candidate_{b\min,i\min}^L=\arg\min_{\tilde{\mathbf{u}}\_candidate_{b,i}^L} J(\tilde{\mathbf{u}}\_candidate_{b,i}^L)$ If $\min(J_{b,i}^L)<\xi$ (for $b=1:r(L)$, and $i=1:k$)

Break the program, and output $\tilde{\mathbf{u}}\_candidate_{b\min,i\min}^{L-1}$ as the kernels parameters for the regression model with $L-1$ -terms.

End if

We select $r(L)$ best elements from all of $\mathbf{u}\_candidate_{b,i}^L$ (for $b=1:r(L-1)$, and $i=1:k$), which can minimize $J_{b,i}^L$'s . The matrixes $\mathbf{u}\_candidate_c^L$ ($c=1:r(L)$) are used to record the $r(L)$ selected elements.

End for

Note 3.

When one let $r(L-1)=k^L$, this algorithm is tree(k). And if $r(L)\equiv s$ for any positive integer L, the algorithm shrinks to be subtree(s, k). If $r(L)\equiv 1$, one can get OLSR with RWBS.

B. Complexity Analysis

However, in the tree(k) algorithm, the number of nodes at each level grows exponentially. In order to obtain a $L-$term regression model, the total number of nodes needed be computed will be $(k^{L+1}-1)/(k-1)-1$. It will cause a heavy computational load. For subtree(s, k) algorithm, only $(L-1)\cdot(k-1)+1$ nodes need to be calculated. So sub-tree algorithm will reduce the computational load of tree algorithm greatly.

IV. SIMULATIONS

In order to show the approximation and detection performance of the proposed algorithm, we ran the experiments on the 1.86GHz notebook with 512MB of RAM, using the Windows XP operating system and employing the MATLAB software. In the simulations, OLSR with RWBS, tree based and subtree based algorithms were compared, all of them with RBF kernels. We selected the parameters in RWBS by cross-validation method. For the 1-dimensional simulations, the parameters of all tree based and subtree based algorithms are set as $Ps=3, NG=20$ and Nb=6 . And for the 2-dimensional simulation, the parameters are $Ps=4, NG=20$ and Nb=10 .

Simulation 1. Components detection

In this simulation, we used the model consisting of three RBF kernels, which is described as (1)

$$f(x)=\exp[-\frac{(x+5)^2}{2\times 1.9^2}]+2\exp[-\frac{(x-4)^2}{2\times 0.4^2}]+4\exp[-\frac{(x-6)^2}{2\times 0.8^2}] \quad (6)$$

A dataset is generated with size $N=160$ by an additive noise process $y_i=f(x_i)+n_i$, where the inputs were uniformly sampled from the domain [-8, 8] and the noise $n_i\sim N(0, 0.1^2)$. Totally 100 times Monte-Carlo trials were performed. Fig. 3 and 4 show the typical performances of OLSR and tree(3), both regression models with only 3 regressors.

Fig.3 (a-c) show the performances produced by each of the three terms (or regressors), that is regressor 1, 2 and 3 respectively. Fig.3 (d) shows the performances of the OLSR with 3 terms. From Fig.3, one can find that, because the first term is not capable of matching one of the waves in the original function properly (see Fig.3 (a)), the third term can not detect any component (see Fig. 3(c)). It is largely due to OLSR is a greedy algorithm, which only seeks the best performance in the current stage, and ignores the affect on the next stage. And the final result of OLSR is bad (see Fig.3 (d)).

On the contrary, because each term of tree (3) can match one of the waves of original function properly, the final model can detect all of the components in equation (6). Tree(3) outperforms OLSR because it has a more global view than OLSR.

This simulation indicates that tree based algorithm is more suitable than OLSR with RWBS to tackle the component detection problem.

Simulation 2. Sparse Representation for ECG Recording

ECG recordings are extremely non-flat signals in time domain, thus the modeling task for ECG is very difficult. Here, we used the first 1.5 second recording, totally 540 samples in “MIT 100” ECG signal [15].

Fig. 5 shows the convergence rates of algorithms with the threshold of training accuracy 0.0020, which indicates that both tree based and subtree based algorithms have much faster convergence rate than OLSR. The modeling performances of algorithms at the 3-th, 5-th and 7-th step are shown in Fig. 6. For sub-tree based and tree based algorithms, both training error and time (second) are also presented in every sub-figure. One can find that subtree based algorithm has a much lower complexity than tree based algorithm. So the tree based algorithm is suitable for the case in which time consumption is not concerned greatly. And subtree based algorithm can be used as a tradeoff between OLSR with RWBS and tree based method.

The simulation shows, for ECG modeling task, that newly proposed algorithms have much faster convergence rates than OLSR. Thus, with a smaller size of regression model, the newly proposed algorithms have much better approximation performance than OLSR.

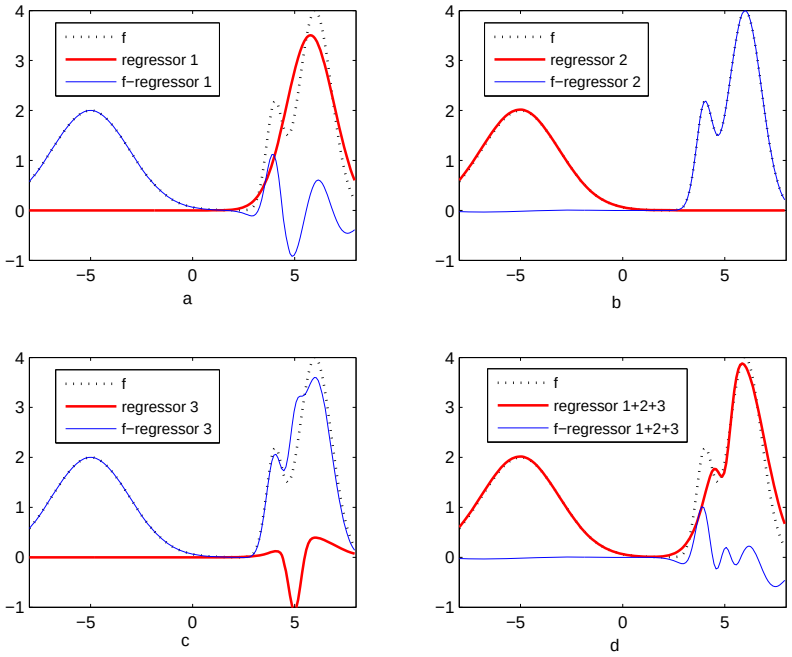


Figure 3. The effect of every regressor of OLSR model with three terms. In Fig.3 (a-c), the dotted line, broad solid line and broad thin line denote the original noise-free function, one of the three regressors of the OLSR model and the difference between them. For Fig.4(d), the thin solid line denotes the model with three terms.

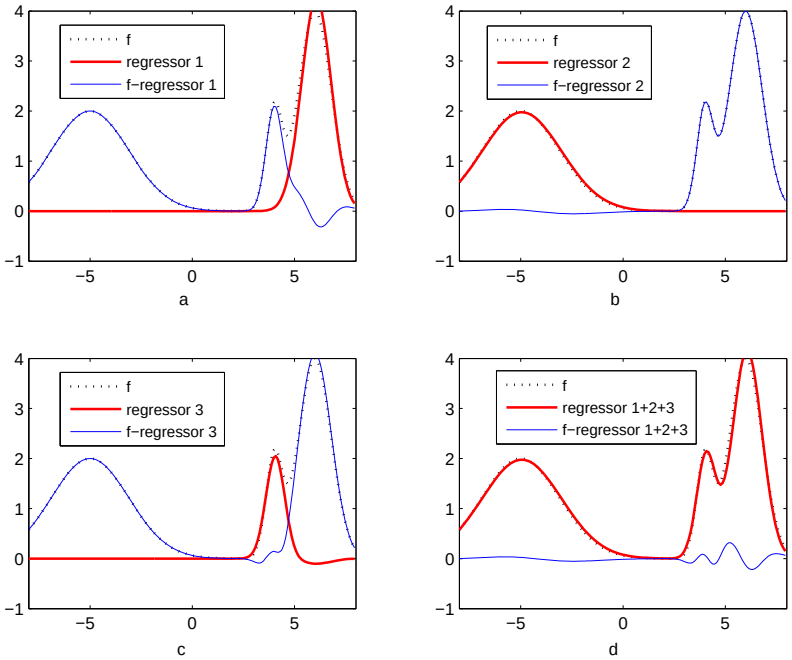


Figure 4. The effect of every regressor of tree(3) model with three terms. For Fig. 4 (a-c), the dotted line, broad solid line and thin solid line denote the original noise-free function, one of the three regressors of the tree(3) model and the difference between them. For Fig.4(d), the thin solid line denotes the model with three terms.

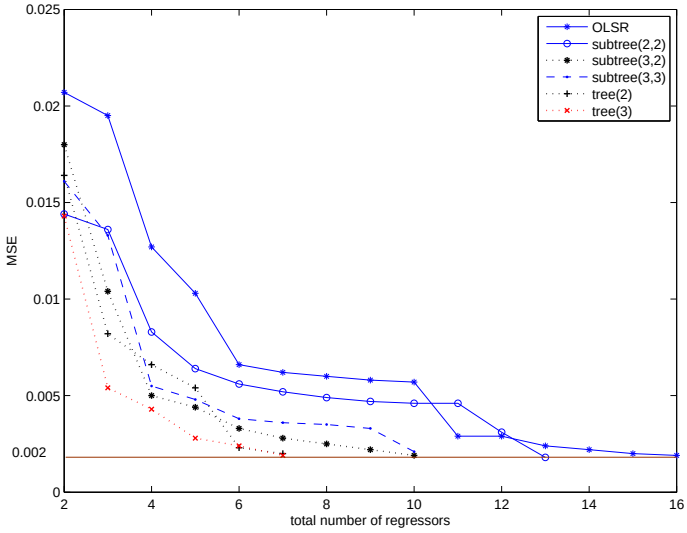


Figure 5. Convergence rates of different algorithms for ECG recording.

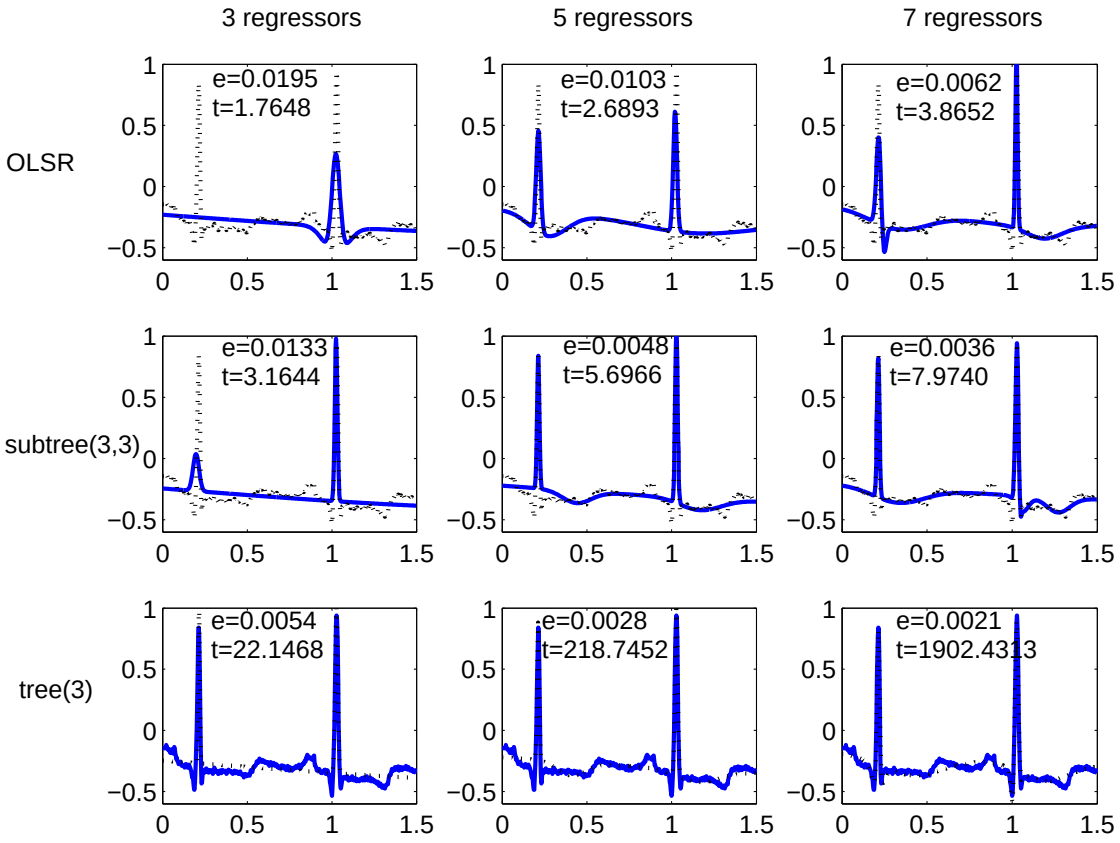


Figure 6. The performances for OLSR, subtree(3,3) and tree(3) algorithms at the different steps. Each algorithm's performance at different step is present in one of the rows.

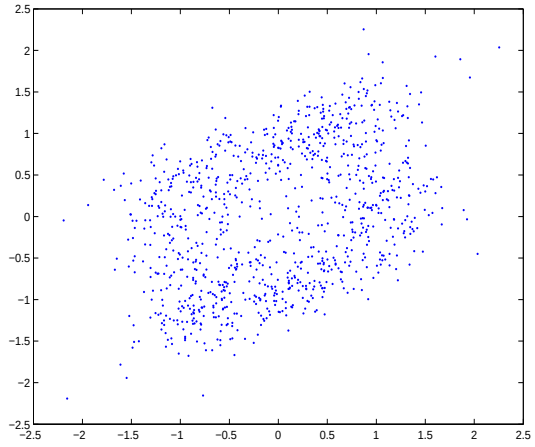


Figure 7. Phase plot of a typical set of noisy training data ($y(0)=y(-1)=0.0$) for the two-dimensional time series modeling problem

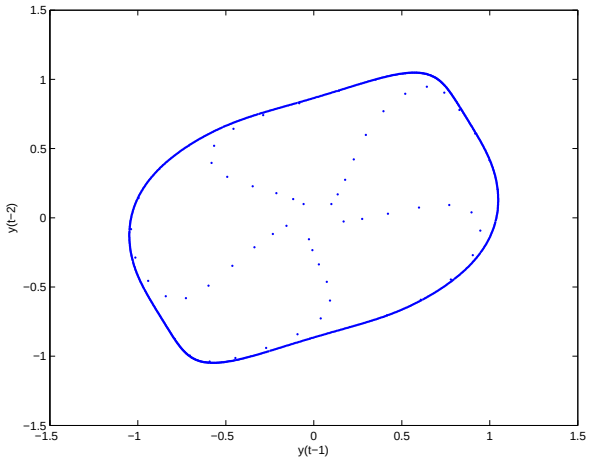


Figure 8. Phase plot of the noise-free two-dimensional time series ($y(0)=y(-1)=0.1$)

Simulation 3. Two- dimensional modeling

This was a two-dimensional simulated nonlinear time series given by

$$y(k)=[0.8-0.5\exp(-y^2(k-1)]y(k-1)$$
$$-[0.3+0.9\exp(-y^2(k-1)]y(k-2)+0.1\sin(\pi y(k-1))+\varepsilon(k)$$

where the noise $\varepsilon(k)$ was Gaussian with zero mean and variance 0.09. One thousand noisy samples were generated given $y(0)=y(-1)=0.0$. This model has been used in [16]. The first 500 data points plotted in Fig.7 were used for training, and the other 500 samples were used for possible cross-validation. The underlying noise-free system

$$y_d(k)=[0.8-0.5\exp(-y^2(k-1)]y(k-1)$$
$$-[0.3+0.9\exp(-y^2(k-1)]y(k-2)+0.1\sin(\pi y(k-1))$$

was shown by 1000 samples given in Fig.8 with $y(0)=y(-1)=0.1$.

Let the input vector $x(k)=[y(k-1), y(k-2)]^T$ and we use OLSR, subtree(2,2), subtree(3,3), subtree(3,2), tree(2) and tree(3) for this 2-dimensional modeling problem.

Let threshold of training error be 0.09, Tab.1 shows the averaged experimental results for 50 times simulations with all tested algorithms and the algorithms mentioned in [16] as well. Fig. 9 shows the modeling performances of different subtree and tree based algorithms at the 5-th step. The simulation shows that both tree and subtree based algorithms can get a better performance in a high-dimensional case than some traditional OLSR algorithms.

V. CONCLUSIONS

In order to a sparse representation, this paper proposes a novel tree based orthogonal least squares regression. Unlike most of the conventional OLSR, the new method keeps the *k* excellent regressors which minimize the modeling MSE, rather than only choose the best one at each iteration. These surviving individuals are used to calculate the *k* new subspaces and corresponding residuals. Because the tree structure search considers the performance not only in the current stage, but also in a global view, this method will lead to a meaningful improvement to the conventional OLSR. Numerical simulations are performed in some signal processing applications, such as component detection, sparse representation for ECG recording, and 2-d time series modeling. In all these simulations, both tree based and subtree based algorithms outperform OLSR with RWBS in convergence rate and accuracy. Because the subtree algorithm avoids the exponentially increasing computational load in tree based algorithm, the former can be regarded as a tradeoff between the latter and OLSR with RWBS.

REFERENCES

[1] A.Smola, "Regression estimation with support vector learning machines," Master's Thesis, Technische University Műnchen, 1996. Available at (http:// www.kernel-machines.org)

[2] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J.Mach. Learning*, pp.211-244,2001.

[3] Z. D. Nian, J. Wang and Y. Zhao, "Non-flat function estimation with a multi-scale support vector regression," *Neurocomputing*, vol. 70, issues 1-3, pp. 420-429, 2006.

[4] S. Chen, S. A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no.5, pp.1873-1896, 1989.

[5] S. Chen, E. S. Chng and K. Alkadhimi, "Regularized orthogonal least squares algorithm for constructing radial basis function networks," *Int. J. Control*, vol.64, no.5, pp.829-837, 1996.

[6] S. Chen, Y. Wu and B. L. Luk, "Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks," *IEEE Trans. Neural Networks*, vol.10, no.5, pp.1239-1243, 1999.

[7] S. Chen, "Locally regularized orthogonal least squares algorithm for the construction of sparse kernel regression models," in Proc. 6th Int. Conf. Signal Processing (Beijing, China), Aug. 26-30, 2002.

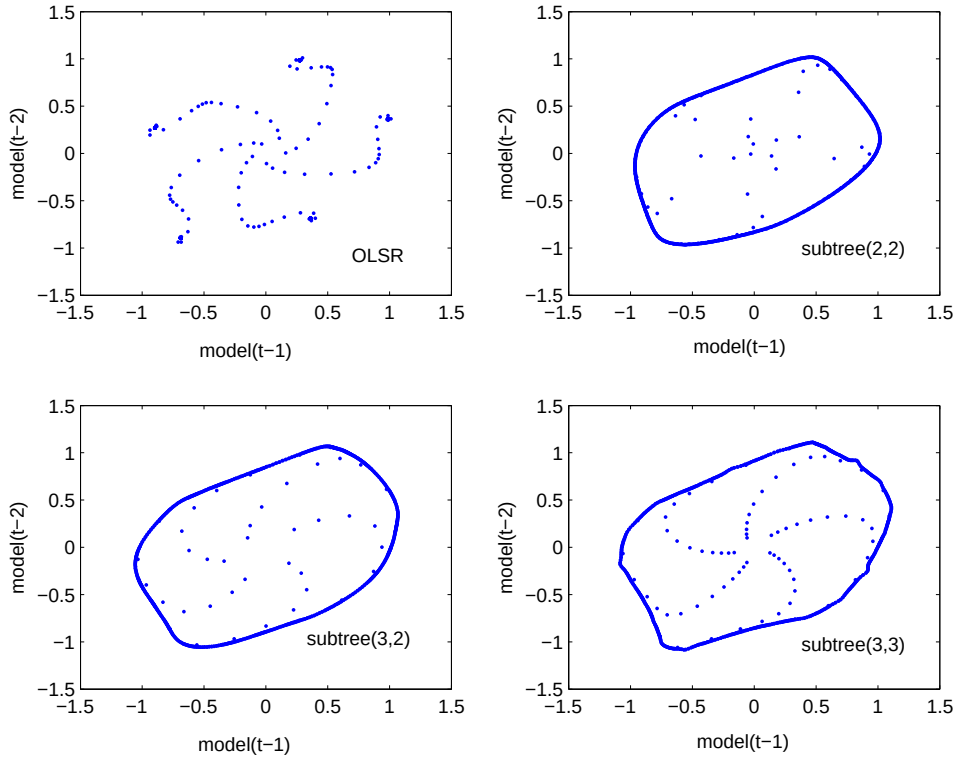


Figure 9. Performances of different algorithms for two-dimensional time series modeling problem at the 5-th step.

Table 1. Performances of different algorithms for two-dimensional modeling. The results of the above 4 rows are cited from [16]

algorithm	MODEL SIZE	MSE
OLS without RWBS	37	0.0881
UROLS	30	0.0911
RVM	19	0.0922
LROLS	18	0.0926
OLSR with RWBS	13.3	0.0903
subtree(2, 2)	7.1	0.0907
subtree(3, 3)	5.2	0.0945
subtree(3, 2)	5.5	0.0967
tree(2)	5.3	0.0970
tree(3)	5.1	0.0964

[8] X. Hong and C. J. Jarris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *IEEE Trans. Neural Networks*, vol.13, no.5 , pp. 1245-1250, 2002.
[9] S. Chen, X. X. Wang, and D.J. Brown, "Orthogonal least squares regression with tunable kernels," *Electronics Letters*, vol. 41 no. 8, 2005.
[10] X. X. Wang, S. Chen, D. Lowe and C.J. Harris, "Sparse support vector regression based on orthogonal forward selection for the generalized kernel model," *Neurocomputing*, vol.70, no.1-3, pp.462-474, 2006.
[11] X. X. Wang, S. Chen, D. Lowe and C.J. Harris, "Parsimonious least squares support vector regression using orthogonal forward selection with the generalized kernel model," *Int. J. Modeling, Identification and Control*, vol.1, no.4, pp.245-256, 2006.
[12] S. F. Cotter, and B.D. Rao, "Application of tree-based searches to matching pursuit," *IEEE. Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3933-3936, 2001.
[13] G.Z. Karabulut, L. Moura, D. Panario, and A. Yongacoglu, "Integrating flexible tree searches to orthogonal matching pursuit algorithm," *IEE ProcVis Image Signal Processing*, vol.153, no.5, pp.538-548, 2006.

[14] S.Chen. X. X. Wang, and C.J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," *IEEE Trans. Syst. Man Cybern. B, Cybern.* vol.35, no.4, pp. 682-693, 2005.
[15] MIT-BIH Arrhythmia Database [Online]. Available: http://www.physionet.org/physio-bank/database/mitda/
[16] S. Chen, "Local regularization assisted orthogonal least regression", *Neurocomputing*, vol. 69, pp559-585, 2006.

ACKNOWLEDGEMENTS

This work was partially supported by NSFC under Grants 11026145, 61071188 and 90920005, by NSF of Hubei Province under Grants 2010CDB04205, 2009CDB077 and by the Fundamental Research Funds for the Central Universities.

Lihua Fu was born in Zhijiang, China, in 1979. She received the Master degree in mathematics from Hubei University. She is currently working toward Ph.D. degree in applied mathematics in China University of Geosciences (CUG). She is also a Lecturer with the School of Mathematics and Physics in CUG. Her research interests include wavelet and its applications in signal processing.

Hongwei Li is with the School of Mathematics and Physics in China University of Geosciences. His research interests include time series analysis and ICA.

Meng Zhang was born in Wuhan, China, in 1977. He received the Master degree in mathematics and the Ph.D. degree in computer science from Wuhan University in 2002 and 2005, respectively. His research interests include signal processing and neural network.

Towards a Syntactic Structural Analysis and an Augmented Transition Explanation: A Comparative Study of the Globally Ambiguous Sentences and Garden Path Sentences

Ping-Fang Yu
School of Liberal Arts, Ludong University, Yantai, Shandong Province, China
Email: yupingfang68@126.com

Jia-Li Du
School of Foreign Languages/School of Literature, Ludong University,Yantai, Shandong Province /Communication University of China, Beijing, China
Email: dujiali68@126.com

***Abstract*—Globally ambiguous sentences can be understood in more than one way and any decoding involved can be understandable. Garden Path (GP) sentences possess a special structural feature with a semantic trigger, which calls for a much heavier cognition burden since only one decoding is reasonable. The present paper explores the data structures of globally ambiguous sentences and GP ones. The augmented transition networks (ATNs) are used to deepen the analyses and comparison of globally ambiguous sentences and GP ones. We come to the conclusion that GP sentences, syntactically and cognitively different from globally ambiguous ones, have more complex structures.**

***Index Terms* -- natural language processing, data structure, augmented transition networks, garden path sentences, ambiguous sentences**

I. INTRODUCTION

Globally ambiguous phenomenon is different from partially ambiguous one to which the Garden Path (GP) phenomenon belongs. Any decoding of globally ambiguous sentences can be accepted by readers while the decoding of GP sentences is involved in the processing breakdown.

The GP sentences occurring at the different levels of human languages have attracted the attention of language researchers, who study the GP sentences from the perspective of grammar [1, 2], semantics [3, 4], pragmatics [5, 6], psychology, information processing[7, 8, 9, 10], and so on. The present paper, on the basis of cognitive linguistics and with the help of the technology of information processing, argues that the key point in decoding the GP sentences is that there occurs a cognitive enlightenment point when the decoding process is stopped, and a kind of structure rearrangement is formed to make the decoding process continue to work smoothly. This kind of semantic tracing back can be explored by way of constructing an appropriate data structure. A data structure is a set of different relations among the various elements in a system, which can roughly be categorized as the set structure, the corresponding linear structure, the multi-matching parallel schema and one-to-many tree diagram. The four afore-cited data structures approximately correspond to the explanation of the pre-grammatical word set, the linear collocation, the many-to-many ambiguity and the semantic tracing back. We'll then explore the different forms of data structures when decoding the GP sentences and probe into the varieties of linguistic structures, with GP sentences included. Besides, a thorough investigation is conducted to explore the grammatical and cognitive difference between the GP sentences and the globally ambiguous ones, aiming to show the structural features of the GP sentences.

II. THE DATA STRUCTURE-BASED ANALYSES OF THE NON-GARDEN-PATH SENTENCES: A COMPARATIVE PERSPECTIVE

Regarding the data structures of the non-GP sentences, there mainly involve these structures, viz. the pre-grammatical word set structure, the grammatical linear structure and the ambiguous schema one.

The pre-grammatical structure means that the words in a set are randomly piled together, without grammatical requirements or semantic compatibility, which is a kind of "word salad" from the psychologists studying how human beings produce languages. Figure 1 shows how the word set structure is like.

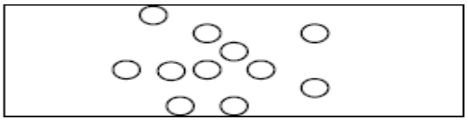


Figure 1. The pre-grammatical word set structure

The next three sentences are the typical examples showing the pre-grammatical word set structure.

Sentence 1: * The new singers the song.

Sentence 2: *The old women the boat.

doi:10.4304/jcp.7.1.196-206

Sentence 3: *The building window the sun.

This kind of word sets is not created together according to the conventionally accepted grammar rules of the target language, so the whole linear construction is ungrammatical, showing a feature of being completely unrelated both grammatically, semantically or pragmatically. For example, "[The new singers] NP + [the song] NP" in Sentence 1, "[The old women] NP + [the boat] NP" in Sentence 2, and "[The building window] NP + [the sun] NP" in Sentence 3 all have the grammatically disconnected linguistic units piled together, none of them showing a shade of semantic compatibility and grammatical acceptance and making no senses at all.

Unlike the random assemblies of the words in the word set structure, the combination of the words governed by the grammatical linear structure is grammatically accepted, and there does exist a one-to-one matching pattern between the words and the conventional subject-predicate structure. The following is the figure concerning the structure of the non-GP sentences of this kind.

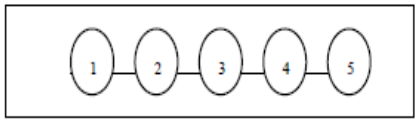


Figure 2. The grammatical linear structure of non-GP Sentences

In the process of syntactic analyses, the elements in the linear structure must be arranged in a sequentially unique way, which means that one and the same element must be syntactically accepted; thus the order of the sentence decoding of the above-mentioned kind has to be processed one by one in order of "1-2-3-4-5". The followings are the sentences of this kind of data structures.

Sentence 4: The new singers record the song.

Sentence 5: The old women sail the boat.

Sentence 6: The building window reflects the sun.

"[The new singers] NP + [record]V + [the song]NP" in Sentence 4, "[The old women] NP + [sail]V + [the boat] NP in Sentence 5, and "[The building window] NP + [reflects]V + [the sun]NP" in Sentence 6 are all functioning properly in the Subject-Verb-Object constructions of the sentences. All the words and the phrases in the sentences have their unique syntactic functions and show a striking feature of a strict combination sequence. This kind of the sentence structure is conventionally regarded as the basic sentence pattern in human languages, which shows no sign of GP phenomenon in our language processing.

Contrary to the grammatical linear structure, the elements in an ambiguous schema structure show a many-to-many grammatical relation, which means that whatever semantic-grammatical corresponding relations we have chosen, a plausible meaning decoding is available. The data structures of the above-mentioned sentences are as follows.

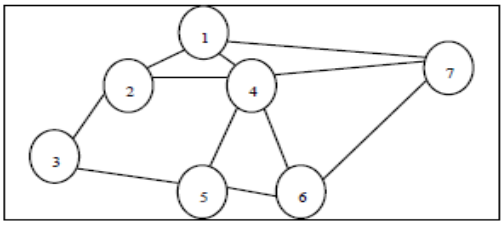


Figure 3. The data structure of ambiguous schema

The cognitive nodes in Figure 3 can be construed in some different ways, which means that no matter what directions the node word moves to, some plausible yet successful textual decoding is available, and consequently various meaning explanations occur. Let's take an example of Sentence 7.

Sentence 7: Failing student looked hard.

There are a number of explanations when we try to decode Sentence 7, and the main reasons of the multiple understandings lie in the polysemous meanings contained in the two key lexemes "falling" and "hard". Grammatically speaking, the word "falling" can be categorized either as an adjective or a gerund; and the word "hard" also has a double grammatical categorization both as an adjective and an adverb. Therefore, the two words with dynamic word classes have potentially provided four different explanations, which are demonstrated separately by way of Sentence 8 to Sentence 11.

Sentence 8: Failing (adj) student looked hard (adj).

Sentence 9: Failing (adj) student looked hard (adv).

Sentence 10: Failing (Grd) student looked hard (adj).

Sentence 11: Failing (Grd) student looked hard (adv).

Sentences 8-11 are to be grammatically analyzed in the following ways.

G={Vn, Vt, S, P}

Vn={S, NP, VP, Adj, Grd, V, N, Adv}

Vt={failing, student, looked, hard }

S=S

P:

S → NP VP	(a)
NP → Adj N	(b)
NP → Grd N	(c)
VP → V Adj	(d)
VP → V Adv	(e)
Adj → {failing, hard}	(f)
Grd → {failing}	(g)
N → {student}	(h)
V → {looked}	(i)
Adv → {hard}	(j)

As shown in the preceding rules, the NP (b) in Line 7 corresponds to the grammatical status set by the word "falling" in Example 8 and Example 9. According to the conventionally accepted grammatical rules, a noun and a verb form a noun phrase; therefore the word "falling" with a word class of adjective can be combined with "student"; thus a noun phrase comes into being. And when the noun phrase co-occurs with the word "hard", which can be tagged either as an adjective or an adverb, two possible decoding procedures, viz. "f-h-b-i-f-d-a" in Sentence 8 and "f-h-b-i-j-e-a" in Sentence 9 are both grammatically and semantically plausible. The two

different decoding procedures result from the selection of the word classes of "hard". To paraphrase the two sentences, Sentence 8 means that "the students having been failed is suffering a lot" and Sentence 9 can be paraphrased as "the student having been failed seems to be working hard". Table I in the following part shows the potential multi-combinations of falling (adj) and "hard (adv)".

TABLE I.
THE POTENTIAL MULTI-COMBINATION OF "FAILING" (ADJ) AND "HARD"(ADV)

	1.	Failing student looked hard			
	2.	Adj student looked hard	(f)		
	3.	Adj N looked hard	(h)		
	4.	NP looked hard	(b)		
	5.	NP V hard	(i)		
6.	NP V Adj	(f)	6'	NP V Adv	(j)
7.	NP VP	(d)	7'	NP VP	(e)
8.	S	(a)	8'	S	(a)

The NP(c) in Line 8 corresponds to the grammatical status set by the word "falling" in Sentence 10 and Sentence 11. It's universally acknowledged that a gerund can be combined with a noun, forming a noun phrase. Therefore the word "falling", tagged as a gerund, can be collocated with the word "hard", which is tagged both as an adjective and an adverb. Then we'll have two sentences with different decoding procedures. To be more specific, the decoding procedure of Sentence 10 is "g-h-c-i-f-d-a", which is paraphrased as "it's no easy job to fail the student". And the decoding procedure of Sentence 11 is "g-h-c-i-j-e-a", which means that "someone is working hard with the aim of failing the student".

TABLE II.
THE POTENTIAL COMBINATION OF "FAILING" (GRD) AND "HARD"(ADV)

	1.	Failing student looked hard			
	2.	Grd student looked hard	(g)		
	3.	Grd N looked hard	(h)		
	4.	NP looked hard	(c)		
	5.	NP V hard	(i)		
6.	NP V Adj	(f)	6'	NP V Adv	(j)
7.	NP VP	(d)	7'	NP VP	(e)
8.	S	(a)	8'	S	(a)

According to the above-mentioned analyses, we know that grammatically the word "falling" shows a double word classes just as Line 11(f) and Line 12(g) have shown to us. Therefore, the generation of the NP, governed by the rules of Line 7(b) and Line 8(c), is ambiguous. Similarly, the word "hard' also has a double word classes shown in Line ll(f) and Line 15(j), which makes the generation of the VP work according to either the rule of Line 9 (d) or that of Line 10 (e). The dual ambiguities of the two words in word classes bring in four different ways of semantic decoding when they are collocated, which is shown in Table III.

TABLE III.
THE POTENTIAL MULTIPLE MATCHES OF "FAILING" AND "HARD" OWING TO THEIR POSSIBLE WORD CLASSES.

	1. Failing student looked hard							
2.	Adj student looked hard	(f)			2''	Grd student looked hard		(g)
3.	Adj N looked hard	(h)			3''	Grd N looked hard		(h)
4.	NP looked hard	(b)			4''	NP looked hard		(c)
5.	NP V hard	(i)			5''	NP V hard		(i)
6.	NP V Adj (f)	6'	NP V Adv (j)		6	NP V Adj (f)	6'	NP V Adv (j)
7.	NP VP (d)	7'	NP VP (e)		7	NP VP (d)	7'	NP VP (e)
8.	S1 (a)	8'	S2 (a)		8	S3 (a)	8'	S4 (a)

It's clearly shown by the preceding Table III that the multiplicity of the lexical-grammatical status of the two collocates entails the non-uniqueness of the sentences generated according to accepted grammatical rules. This kind of ambiguity is to be shown with the help of a many-to-many schema structure, which is demonstrated in Figure 4.

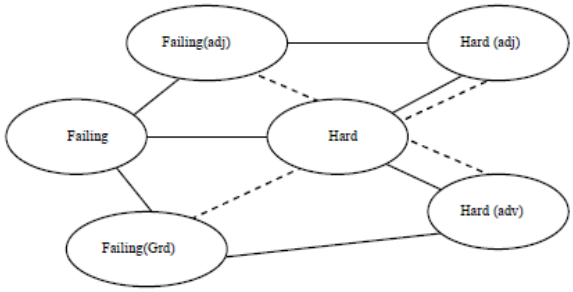


Figure 4. The data structure of the sentence "Failing student looked hard"

III. ANALYSES OF THE DATA STRUCTURE OF GARDEN-PATH SENTENCES

The non-GP sentences governed by either the word set structure, the grammatical linear structure or the ambiguous schema structure have a defining characteristic of being mono-directional. Unlike the non-Garden-Path ones, the GP sentences show a distinguishing and recurring feature of being forced to turn back to the original starting point to find a new way out in the decoding process, which turns out to be a tree diagram regarding its data structure. The tree diagrams of the GP sentences mean that there does exist a one-to-many corresponding relation among the elements in the structure under discussion. When one of the relations is by default interpreted as a cognitive prototype, it will automatically become the root of the whole decoding process; then extend along the seemingly plausible decoding route until a sudden semantic breakdown calls for a prompt return to the starting point. Afterwards, it will move along a next node, until all the elements in the structure go smoothly in the decoding process. The structure of sentences of this kind is shown in Figure 5.

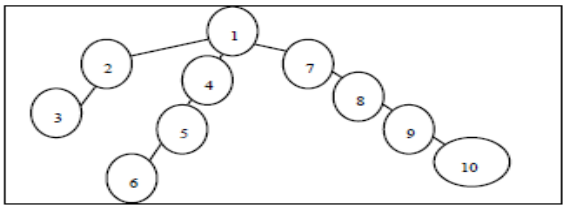


Figure 5. The tree diagram of the Garden-Path sentences

Suppose the root element in the above-mentioned figure corresponds to three different ways of semantic cognition, and the appropriate way of the decoding of the target sentence is No.10; then we'll have the possible order of decoding as "1-2-3-3-2-1-4-5-6-6-5-4-1-7-8-9-10". It can be easily found that a kind of semantic returning is necessarily entailed in the decoding of the GP sentence. The simultaneous appearances of the two decoding routes viz. "3-2-1" and "6-5-4-1" demonstrate that minimally two tracing-backs are included in the process of sentence understanding, which demonstrates that the double negation turns out to be an indispensable part of the sentence decoding and it can be statistically verifiable. We'll have another example.

Sentence 12: The new record the song.

Since the word "record" can be tagged either as a noun or a verb, the proper understanding of sentence 12 can be achieved by way of the context-free grammar, which is demonstrated in the following part.

G={Vn, Vt, S, P}
Vn={S, NP, VP, Det, Adj, V, N}
Vt={the, new, record, song}
S=S
P:

S → NP VP	(a)
NP → Det Adj	(b)
NP → Det N	(c)
NP → Det Adj N	(d)
VP → V NP	(e)
Det → {the}	(f)
Adj → {new}	(g)
N → {record, song}	(h)
V → {record}	(i)

Processing Procedure (Bottom-Up)

1: The new record the song	
2: Det new record the song	(f)
3: Det Adj record the song	(g)
4: Det Adj N the song	(h)
5: NP the song	(d)
6: NP Det song	(f)
7: NP Det N	(h)
8: NP NP	(c)
9: Det Adj record the song	(g)
10: NP record the song	(b)
11: NP V the song	(i)
12: NP V Det song	(f)
13: NP V Det N	(h)
14: NP V NP	(c)
15: NP VP	(e)
16: S	(a)
SUCCESS	

Judged by the afore-cited theoretical framework of context-free grammar, it's clear that the decoding of Sentence 12 involves a kind of temporary halting and the sequent on-time returning because of a sudden semantic breakdown. According to the tree diagram of Figure 6, the correct sequence of the decoding process is as follows: 1-2(f)-3(g)-4(h)-5(d)-6(f)-7(h)-8(c)-8(c)-7(h)-6(f)-5(d)-4(h)-3 (g)-9(g)-10(b)-11(i)-12(f)-13(h)-14(c)-15(e)-16(a). Owing to the fact that Rules 3(g) - 9(g) are concerned with the decisions that whether the node word "record" should be tagged as a noun or as a verb, the word "record" thus becomes the source of the semantic tracing-back, and brings in two explanations of the target sentences. One decoding procedure is "3(g)-4(h)-5(d)-6(f)-7(h)-8(c)" with the word "record" being a noun; and the other decoding is "9(g)–10(b) –11(i) –12(f) –13(h) –14(c) –15(e) –16(a)" with the word "record" being tagged as a verb. However, the above-mentioned sentence in which "record" is regarded as a noun is not grammatically accepted, and then a new cognition enlightenment point appears: the decoding process then begins to trace back from 8(C) to 3(g), and a kind of GP phenomenon "8 (c) – 7 (h) – 6 (f) – 5 (d) – 4 (h) -3 (g)" comes into being. To be more specific, the word "record" in the sentence has a dual role: grammatically, the word "record" is always tagged as a noun by default reasoning; while functionally the word "record" must be tagged as a verb to make the whole decoding process go smoothly. And the first dead end in the information processing leads to the second semantic tracing back. The tree diagram of this decoding process of Sentence 12 is shown below:

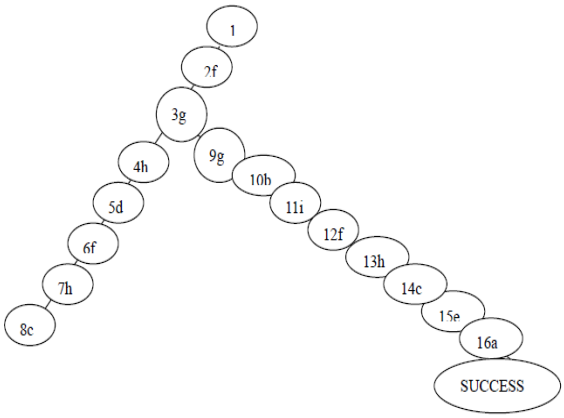


Figure 6. The data structure of "The new record the song"

IV. The Application of Augmented Transition Networks

Augmented Transition Networks (hereafter abbreviated to ATNs) are applied in the computational linguistics to demonstrate the process of the state transitions in NLP[11, 12, 13]. The networks have a main net and some subsidiary ones, all of which are interconnected and interchangeable, allowing the linguistic segments to semantically and syntactically match with other collocates in the main net until they are successfully decoded. ATNs clearly show the hierarchical features of the decoding processes of the sentences with complicated syntactic relations. On basis of the ATNs theory, the following part of the paper endeavors to analyze the decoding processes of the ambiguous sentences and the GP sentences, by which a comparison is made between the two kinds of sentences.

The ambiguous sentences, because of their phonetically, semantically or grammatically various explanations, potentially convey at least two different meanings. Therefore the decoding of the ambiguous sentences necessarily involves a kind of semantic selection. Metaphorically speaking, the decoding of the ambiguous sentences, necessarily involving some semantic options, is like a road with several motorways,

and the motorways are parallel with each other and each of them can take you to the place you want.

Unlike the one-road-with-several-driveway ambiguous sentences, the GP sentences are actually a one-way street. Metaphorically speaking, exploring GP sentences is the same as the fact that someone who is strolling along a zigzagging garden path suddenly finds that the path is a dead end and what he or she can do is to turn around and walk along the same route to the original location just to find another way out. Unlike the multi-explanations of the ambiguous sentences, the GP sentences are to be decoded only in one and the same way. That is to say, in the cognitive decoding process of GP sentences, there will be a semantic stoppage and a sequent backtracking to make the decoding go continuously. This kind of sudden insight will certainly burden and lengthen the decoding process, while it simultaneously develops the decoding capability in a leaping way, which has been empirically tested by scholars from different perspectives [14, 15, 16, 17, 18].

The comparison between ambiguous sentences and the GP sentences has been conducted by the scholars from the perspectives of syntax [19], pragmatics [20], psychology [21], computational science [22] and cognitive science [23, 24, 25, 26]. The present paper, on basis of Kempen's ATN theory, endeavors to compare the ambiguous sentences and GP Phenomena with the aim of exploring the semantic trigger systems contained respectively in ambiguity and GP sentences.

V. The Framework of Kempen's Syntactic Processors

In 1996, G. Kempen published his famous article titled *Computational models of syntactic processing in human language comprehension*. In this article, Kempen discusses the basic framework of syntactic processing machine as well as explores both the ambiguous sentences and GP sentences from the perspective of ATNs.

Kempen argues that the framework of a syntactic processor minimally contains five parts, viz. an input buffer, a syntactic proceesor and working memory, lexicon and grammar, the conceptual knowledge and the syntactic structure. Kempen's model is shown in Figure 7.

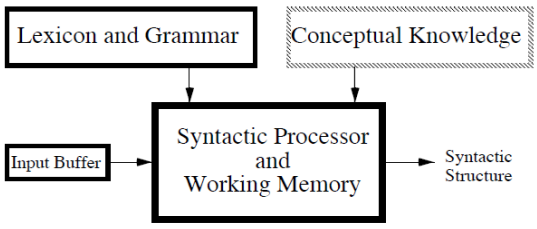


Figure 7. Kempen's (1996) model of syntactic processing procedures

Figure 7 shows that the word strings are firstly stored in the input buffer, and the lexical, morphological, syntactical and conceptual information is fully integrated in the syntactic processor, forming either a complete or a partial syntactic structure. Kempen holds that the detailed explanation in the information processing is varied according to the corresponding models, which thus forms an interactive model. In the syntactic processing of human languages, grammar, lexicon and most of the conceptual knowledge are stored in our long-time memory, which has laid a solid foundation of the application of ATNs in language processing.

ATN was first put forward by Kaplan (1972) and was mainly used to demonstrate the language processing procedures in a series of states. The different states in language processing are demonstrated with labelled nodes and the transition is shown with directed and labelled arcs. The arcs, howerver, are further labelled with some particular syntactic rules, and the transition direction is shown with an arrow. Therefore the whole decoding process is clearly explained. With the help of the demonstration of the states involved in the decoding process, together with the ATNs, Kempen probes into the discrepancies between the ambiguous sentences and the GP sentences. See the following three examples.

Sentence 13: The student read the letter to Chrysanne.

Sentence 14: The student read the letter to Chrysanne fainted.

Sentence 15: The student who was read the letter to Chrysanne fainted.

The three sentences cited above are both semantically and syntactically different. Sentence 13 can be paraphrased as "a student read a letter to a woman named Chrysanne". This kind of decoding is cognitively economical, for there is less structural nodes. Besides, Sentence 13 can also be paraphrased as "a student read a letter, but the letter was meant for Chrysanne". The latter decoding is also syntactically correct. It's safe to say that Sentence 13 is syntactically ambiguous.

Sentence 14 is a typical GP sentence. Before the occurrence of the past participle "fainted", the syntactic pattern of "[[The student] NP [[read] $V_{transitive}$ [the letter]NP[to Chrysanne]PP]VP]S", which accords with the Sausage Machine theory, has come into being. But the participation of the following part has the former syntacitc pattern changed, and forms a new pattern of "[[[The student] NP [[read] $V_{passive}$ [[the letter] NP [to Chrysanne] PP] NP] RRC] NP [fainted] VP] S". This kind of change causes a cognitive insight point and consequently the GP phenomenon occurs.

Sentence 15 is a sentence of commonly used type. The short working memory of the decoder is greatly affected by the word length when decoding the phrase "to Chrysanne", and accords with the Right Association Principle of sentence decoding. Therefore, the syntactic pattern adopted in the process is "[[read] $V_{transitive}$ [[the letter] NP [to Chrysanne] PP] NP] VP" (i.e. "V_{tr}+NP" Pattern), instead of the pattern "[[read] $V_{transitive}$ [the letter] NP [to Chrysanne]PP]VP (i.e. "V_{tr}+NP + PP" Pattern), and there're no ambiguities or GP phenomenon occuring in Sentence 15. As to the details, see Figure 8 and Figure 9.

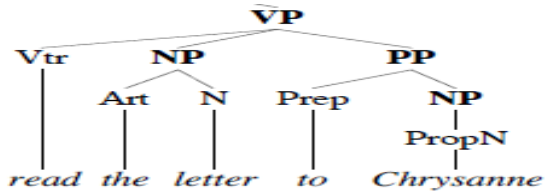


Figure 8. The node pattern of "Vtr+NP＋PP"

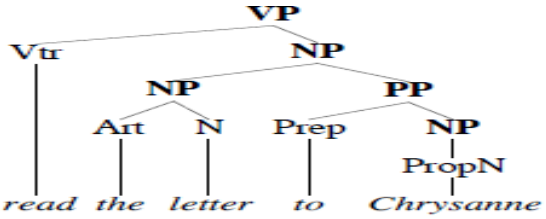


Figure 9. The node pattern of "Vtr+NP"

VI. THE ANALYSES OF THE AUGMENTED TRANSITIONS OF MULTI-STATES IN THE SYNTACTIC PROCESSING

G. Kempen introduces the context-free grammar into his procedure, with the aim of improving the program-based language processing with the help of the ATNs. Figure 10 shows us how the context–free grammar works in Sentence 13 and Sentence 14.

G={Vn, Vt, S, P}

Vn={S, NP, RRC, VP, PP, V_{intr}, V_{tr}, V_{pass}, Art, N, PropN, Prep}

Vt={ the, student, read, letter, to, Chrysanne, fainted}

S=S

P:

1. S → NP VP	6. RRC → Vpass NP	11. PP → Prep NP	16. N → letter
2. NP → Art N	7. VP → Vintr	12. Vintr → fainted	17. N → student
3. NP → PropN	8. VP → Vintr PP	13. Vtr → read	18. PropN → Chrysanne
4. NP → NP PP	9. VP → Vtr NP	14. Vpass → read	19. Prep → to
5. NP → NP RRC	10. VP → Vtr NP PP	15. Art → the	

/* RRC(reduced relative clause) */

Figure 10. The CFG used in the decoding process of Sentences 13 & 14

A. The State Demonstration of the Main Networks

The procedural decoding of the ATNs of Sentence 13 and Sentence 14 can be explained with the help of the following 5 figures, viz. from Figure 11 to Figure 15. Figure 11 shows us what the states of the output of the main net S are like.

Figure 11. The states involved in the output of the main net of "S"

Figure 11 is comprised of four parts, viz., the labelled nodes, the labelling arcs, the denoting lines and the syntactic rules concerned. Labelled nodes "S_0", "S_1" and "S_2" respectively stand for the initial state, the middle state and the output one. The labelling arcs and the denoting lines in Figure 11 converge into the 3 right-directing lines, demonstrating the transition directions of the decoding. The syntactic rules "(SEEK) NP", "(SEEK) VP" and "(SEND) S" are used to denote the grammatical functions. It's clear that Figure 11 shows us the syntactic analyses of "S →NP VP".

B. The State Demonstration of the VP Subsidiary Net

Figure 12 in the following part shows us the demonstration of the VP subsidiary net.

In Figure 12, the labelling nodes "VP_0", "VP_1", "VP_2" and "VP_3" respectively stand for the different states of the decoding process. There are three labelling arcs and three denoting lines. Besides, six syntactic rules, viz. "CAT V_{tr}, SEEK NP, CAT V_{intr}, SEEK PP, JUMP, SEND VP" are involved in the process. In terms of the syntactic rules, "CAT V_{tr}" means the node word is a transitive verb; "SEEK NP" means that a transitive verb needs a noun phrase, and the corresponding syntactic rule is "VP → V_{tr} NP". "CAT V_{intr}" means an intransitive verb is involved, and its syntactic rule is "VP →V_{intr}". "SEEK PP" means that a new verbal phrase is to occur according to the rule "VP→ V_{tr} NP PP" or "VP → V_{intr} PP". "JUMP" is used to show that the newly-formed verbal phrase can directly pass by the matching with a preposition phrase and leap to the output stage. "SEND VP" tells us that the ultimate output is a verbal phrase. Theoretically the "VP_3" here can have 4 results. The first one is "VP→ V_{tr} NP PP" (as the case in "[[read] V_{tr} [the letter] NP [to Chrysanne] PP]VP"), and its state curve is "4—5—7—9"; The second result is "VP → V_{intr} PP" (as the case in "[[read] V_{intr} [to Chrysanne] PP]VP, and the state curve is "6—7—9". The third one is "VP→ V_{tr} NP" (as the case in "[[read] V_{tr} [[the letter] NP [to Chrysanne] PP]NP]VP") with a state curve "4—5—8—9". And the last result is "VP → V_{intr}"(as the case in "[[fainted] V_{intr}]VP") , with a state curve "6—8—9".

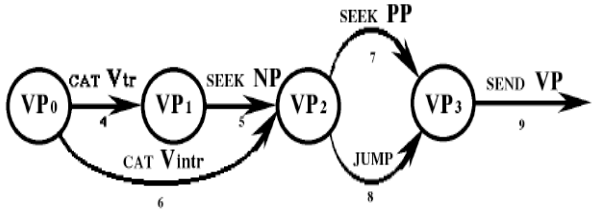


Figure 12. The state demonstration of the output subsidiary net of "VP"

The first and the third models respectively correspond with the state transition model of the two different translations of the verbal phrase "read a letter to Chrysanne", containing two different state curves "4—5 — 7 — 9" and "4 — 5 — 8 — 9", verifying that the ambiguous sentences don't cause backtrackings in the decoding process. The fourth result corresponds with the transition states of "VP →V_{intr}" in Sentence 14, while the

second result can't be applied in the analyses of Sentence 13 and Sentence 14. The details are shown in Table IV.

TABLE IV.
THE THEORETICAL MODEL OF GENERATING STATES OF THE SUBSIDIARY NET OF "VP3"

No.	Program	Trace	Example	Relevance
1	VP→ V_{tr} NP PP	4-5-7-9	[[read] V_{tr} [the letter] NP [to Chrysanne] PP]V	Yes
2	VP → V_{intr} PP	6-7-9	[[read] V_{intr} [to Chrysanne] PP]VP	No
3	VP→ V_{tr} NP	4-5-8-9	[[read] V_{tr} [[the letter] NP [to Chrysanne] PP]NP]VP	Yes
4	VP → V_{intr}	6-8-9	[[fainted] V_{intr}]VP	Yes

C. The State Demonstration of the NP Subsidiary Net

The figure below tells us how the state demonstration of the output subsidiary NP net works.

The labelling nodes "NP_0", "NP_1" and "NP_2" demonstrate the different decoding stages. There are three labelling arcs "10", "14" and "15", together with three denoting lines "11", "12" and "13". Besides, there involve six syntactic rules, viz. "CAT PropN", "CAT Art", "CAT N", "SEND NP", "SEEK PP" and "SEEK RRC".

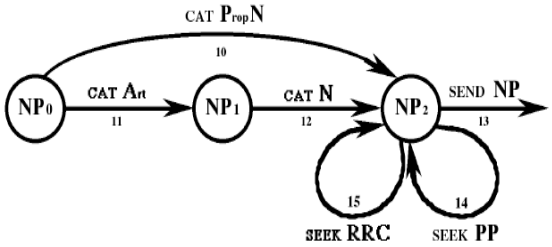


Figure 13. The state demonstration of the output subsidiary net of "NP"

The code "CAT PropN" means that the grammatical category of the node word is a proun. According to the generating rule "NP →Prop N", the state curve of the pronoun is "10-13", viz. "[[Chrysanne] PropN] NP". "CAT Art" and "CAT N" stand for "article" and "noun" respectively. According to the rule "NP →Art N", a new noun phrase comes into being, with a state curve of " 11-12". Code "SEEK PP" means that a prepositional phrase is to be searched to have a grammatical match. When NP2 comes into being, a new noun phrase can be produced. Theoretically, we'll have two results here. The first is the syntactic pattern of "[[[Chrysanne] Prop N] NP [on the park] PP]NP" with a state curve "10-14-13" according to the syntactic rule "NP →Prop N", "PP→ Prep NP" and "NP →NP PP". The second one is a syntactic pattern "[[the letter]NP[to Chrysanne]PP]NP" with a state curve of " 11-12-14-13" according to the rules "NP →Art N", "PP→ Prep NP" and "NP →NP PP". Practically, only the second pattern is applicable in the decoding process of Sentence 13 and 14, and the first pattern can't work because there's no prop noun as the subject in the two sentences.

"SEEK RRC" means a simplified sentence, i.e. reduced relative clause (RRC), is needed to match a new phrase. Theoretically speaking, four syntactic states can be produced. (1) The syntactic rules "NP →PropN", "RRC→ V_{pass} NP" and "NP→ NP RRC" bring in a pattern "[[[Mike] PropN] NP [[read] V_{pass} [the letter] NP] RRC] NP fainted" with a state curve "10—15—13".(2) The syntactic rules "NP →Prop N", "RRC→ V_{pass} NP", "NP→ NP RRC" and "NP →NP PP" form a syntactic pattern "[[[Mike] PropN] NP[[read] V_{pass} [[the letter] NP [to Chrysanne]PP]NP]RRC]NP fainted" with a state curve "10—14—15—13". (3) A new syntactic pattern "[[The student]NP[[read] V_{pass} [[the letter]NP[to Chrysanne]PP]NP]RRC]NP fainted" with a state curve "11—12—14—15—13" is formed according to the rules "NP →Art N", "RRC→ V_{pass} NP", "NP→ NP RRC" and "NP →NP PP".(4)According to the rules "NP →Art N", "RRC→ V_{pass} NP" and "NP→NP RRC", a syntactic model "[[The student]NP [[read] V_{pass} [the letter]NP] RRC]NPfainted" with a state curve "11—12—15—13" comes into being. Among the above-cited syntactic patterns, only the third one can lead to a successful decoding of Sentence 14.

Considering the analyses of the transition states shown in Figure 13, we know that theoretically there are eight forms concerning the output of NP2, while practically only four of them are plausible. See Table V.

TABLE V.
THEORETICAL STATE PATTERN OF THE SUBSIDIARY NET OF "NP2"

No.	Program	Trace	Example	Relevance
1	NP →PropN	10-13	[[Chrysanne] PropN]NP	Yes
2	NP →Art N	11-12-13	[[The]Art[student] N]NP	Yes
3	NP →PropN PP→ Prep NP NP →NP PP	10-14-13	[[[Chrysanne]PropN]NP[on the park]PP]NP	No
4	NP →Art N PP→ Prep NP NP →NP PP	11-12-14-13	[[the letter]NP[to Chrysanne]PP]NP	Yes
5	NP →PropN RRC→ V_{pass} NP NP→ NP RRC	10-15-13	[[[Mike] PropN]NP [[read] V_{pass} [the letter]NP]RRC]NP fainted	No
6	NP →PropN RRC→ V_{pass} NP NP→ NP RRC NP →NP PP	10-14-15-13	[[[Mike] PropN]]NP[[read] V_{pass} [[the letter]NP [to Chrysanne]PP]NP]RRC]NP fainted	No
7	NP →Art N RRC→ V_{pass} NP NP→ NP RRC NP →NP PP	11-12-14-15-13	[[The student]NP[[read] V_{pass} [[the letter]NP[to Chrysanne]PP]NP]RRC]NP fainted	Yes
8	NP →Art N RRC→ V_{pass} NP NP→ NP RRC	11-12-15-13	[[The student]NP [[read] V_{pass} [the letter]NP] RRC]NP fainted	No

D. *The State Demonstration of Other Subsidiary Nets besides the NP and VP*

Figure 14 and Figure 15 below are the state demonstration of the output subsidiary PP net and RRC net.

Figure 14. The state demonstration of the output subsidiary net of "PP"

According to the rule "PP→ Prep NP" and "NP → Prop N", the decoding pattern "[[to] Prep [[Chrysanne] Prop N] NP] PP"is produced.

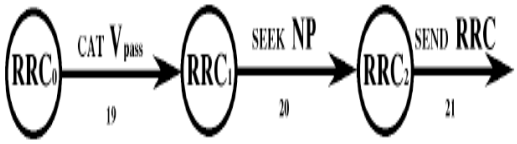


Figure 15. The state demonstration of the output subsidiary net of "RRC"

According to the grammatical rules "RRC→ V_{pass} NP", "NP →Art N" and "NP →NP PP", a new syntactic pattern "[[read] V_{pass} [[[the] Art [letter] N] NP [to Chrysanne] PP] NP] RRC" comes into being。

With the successful decoding of all the subsidiary nets, viz. the VP subsidiary net, the NP net, the PP net, the RRC net, the main net S, the target sentence is to be encoded and decoded thoroughly.

VII. ANALYSES OF THE "PUSH-POP PATTERNS" IN A MULTI-STATE TRANSITION SYSTEM

Involved in the production of Sentence 13 and 14 are three kinds of "push-pop" patterns: the "Vtr+NP＋PP" pattern, the "Vtr+NP" pattern and the GP sentence pattern. Among them, the "Vtr+NP+PP" pattern is the most cognitively economical; the "Vtr+NP" pattern is more complex, with a much higher degree of pushing-down. The GPP pattern is the most cognitively complex and has a most complicated transition state. According to the analyses of Figures 11-15, these different patterns can be shown clearly.

A. *The "Push-Pop" Pattern in the "Vtr+NP＋PP" Construction*

The state curve of Sentence 13 is as follows: Start－1(S_0)－11(NP_0)－12(NP_1)－13(NP_2)－1(S_0)－2(S_1)－4(VP_0)－5 (VP_1)－11(NP_0)－12(NP_1)－13(NP_2)－5(VP_1)－7(VP_2)－16(PP_0)－17(PP_1)－18(PP_2)－7(VP_2)－9(VP_3)－2(S_1)－3(S_2)－success. Among the decoding procedures, a number of pushing-downs and poping-ups are involved. The pushing-downs mean the matching pattern from the main net to the subsidiary nets concerned. Conversely, the poping-ups refer to the stage from the subsidiary net to the main one.

The first pushing-down and poping-up is connected with the matching partner of the noun phrase, viz. " Start －1(S0)－11(NP0)－12(NP1)－13(NP2)－1(S0)" . When S0 is transited to S1, a noun phrase is seeked to be matched grammatically, and then the main net pushes down to the subsidiary NP net, selecting an appropriate collocate from the eight potential noun phrases. The NP "the student" , in line with the pattern "11(NP0)－12(NP1)－13(NP2)", is popped up to the main S net.

The second push-pop occurs in the process of the verbal phrase's matching and the procedure is "2(S1)－4(VP0)－5 (VP1)－11(NP0)－12(NP1)－13(NP2)－5(VP1)－7(VP2)－16(PP0)－17(PP1)－18(PP2)－7(VP2)－9(VP3)－2(S1)－3(S_2)". This kind of push-pop is much more complicated, involving one first-level push-pop, and two second-level push-pops.

The first-level pushing down occurs during the stage of "2(S1)－4(VP0)－5 (VP1)". The first second-level push-pop comes into being in the process of "11(NP0)－12(NP1)－13(NP2)－5(VP1)". And the second second-level push-pop occurs at the stage of "7(VP2)－16(PP0)－17(PP1)－18(PP2)－7(VP2)". The output procedure is in the stage "3(S_2)－success". According to the syntactic rule "S →NP VP", the decoding of Sentence 13 is completed. One first-level and two second-level push-pops are involved. Figure 16 shows the "push-pop" pattern of the "Vtr+NP＋PP" construction, in which the upward arrow stands for the poping-up, and the downward arrow refers to the pushing-down and the linear arrow means the directions.

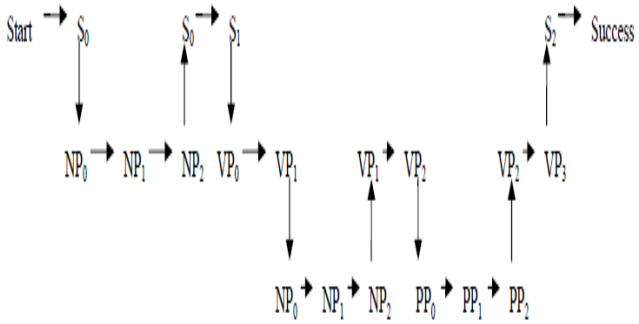


Figure 16. The augmented transition net of "V_{tr}+NP＋PP"

B. *The "Push-Pop" Pattern in "Vtr+NP"*

Another state curve of the decoding of Sentence 13 is as follows: "Start－1(S_0)－11(NP_0)－12(NP_1)－13(NP_2)－1(S_0)－2(S_1)－4(VP_0)－5 (VP_1)－11(NP_0)－12(NP_1)－13(NP_2)－14(PP)－16(PP_0)－17(PP_1)－18(PP_2)－14(PP)－5(VP_1)－8(VP_2)－9(VP_3)－2(S_1)－3(S_2)－Success". This kind of decoding pattern takes up more cognitive resource than the frist pattern, having two first-level push-pops, one second-level push-pop and one third-level push-pop. Because three levels of push-pop are involved, the whole decoding system is more complicated, which empirically verifies the assumption

that the "V_{tr}+NP + PP" pattern takes up much less resource than the "V_{tr}+NP" pattern, which is demonstrated in Figure 17.

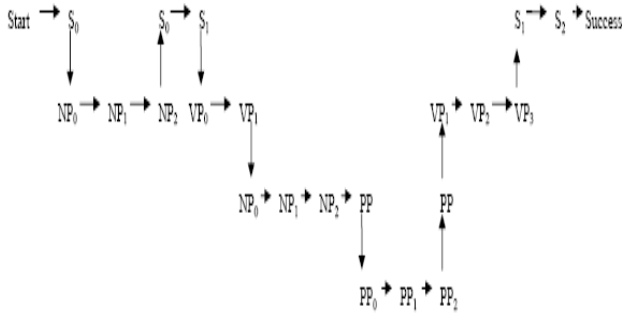


Figure 17. The "push-pop" pattern in the "Vtr+NP" construction

We can find that as an ambiguous sentence, Sentence 13 can be decoded by way of two constructions of "V_{tr}+NP + PP" and "V_{tr}+NP". Provided with the same amount of working memory, the "V_{tr}+NP+PP" construction is preferred because it's much more cognitively economical. Besides, the two constructions don't cause a backtracking.

C. The "Push-Pop" Pattern in the Garden Path Sentences

The decoding system of Sentence 14 is much more complicated because a backtracking occurs in the decoding process and two kinds of decoding procedures are involved and coincided, with a much heavier cognitive burden. According to the Minimal Attachment Principle, the most cognitively economical pattern "V_{tr}+NP + PP" is adopted. Sentence 14, however, complies with the Right Association Principle, and the most complicatd syntactic pattern "V_{tr}+NP" is chosen. The following 3 figures show the difference of the corresponding tree diagrams.

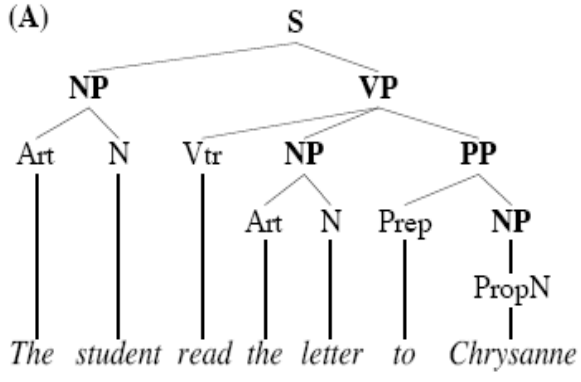


Figure 18. The tree diagram of the ambiguous " Vtr+NP+PP" construction governed by the Minimal Attachment Principle

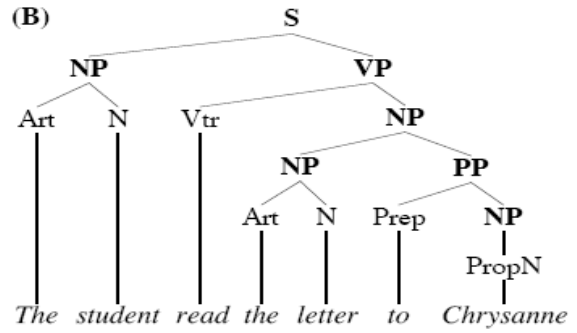


Figure 19. The tree diagram of the ambiguous "Vtr+NP" construction governed by the Right Association Principle

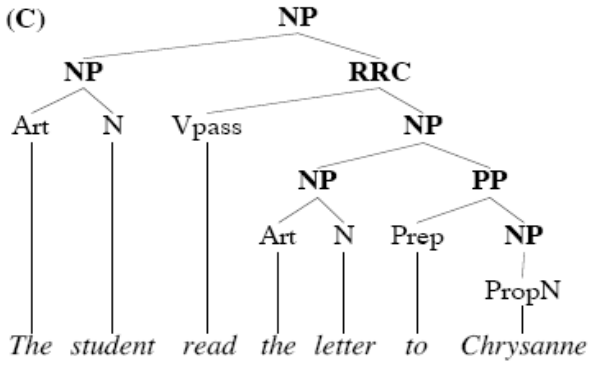


Figure 20. The tree diagram of the Garden Path sentences

The variation of the PP modifiers is the main reason which causes the GP phenomenon. Besides, the change of the predicate verb can also produce GP phenomenon. Before the occurrence of "fainted", the predicate of Sentence 14 is assumed to be "read" and the corresponding syntactic rule is "VP → V_{tr} NP". All is changed after the appearance of "fainted". The word "read", according to the "RRC → V_{pass} NP" rule, is downgraded as the verb in the clause, and " fainted" , however, is upgraded as the main verb of the whole sentence. The readjustment of the main verb also causes the GP phenomenon.

Sentence 14, being one of the typical GP sentences, entails a modifier variation and a verb change, and its decoding state curve is divided into 4 phases.

The first phase is the transition state before the occurrence of the word "fainted", something like the "Vtr+NP + PP" pattern in line with the Minimal Attachment Principle. The curve is "Start − 1(S_0) − 11(NP_0) − 12(NP_1) − 13(NP_2) − 1(S_0) − 2(S_1) − 4(VP_0) − 5(VP_1) − 11(NP_0) − 12(NP_1) − 13(NP_2) − 5(VP_1) − 7(VP_2) − 16(PP_0) − 17(PP_1) − 18(PP_2) − 7(VP_2) − 9(VP_3) − 2(S_1) − 3(S_2) − backtracking".

The second phase is from the restart of the decoding to the match of the NP. The state curve of the decoding is "Restart − 1(S_0) − 11(NP_0) − 12(NP_1) − 13(NP_2) − 19(RRC_0) − 20(RRC_1) − 11(NP_0) − 12(NP_1) − 13(NP_2)".

The third phase is the backtracking after the occurrence of the prepositional phrase. Its decoding state curve is as follows: "20(RRC_1) − 21(RRC_2) − 15(RRC) − 13(NP_2) − 1(S_0) − 2(S_1) − backtracking". The

backtracking in the third stage is the second reentry, which is represented with dotted lines.

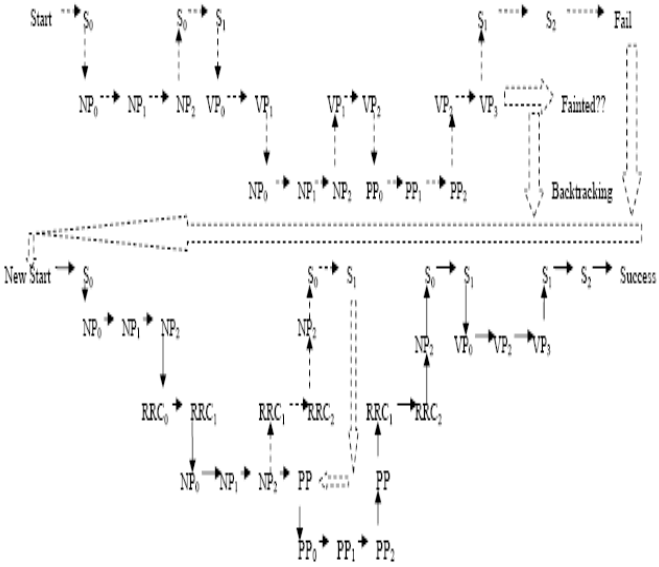


Figure 21. The "push-pop" pattern of Garden Path sentences

The fourth stage is from the end of the backtracking to the success of the decoding. After a second reentry, the grammatical categories of the word "read" and the phrase "to Chrysanne" are determined, which gives a rapid processing of Sentence 14. The state curve of the "push-pop" concerning the stage in discussion is as follows: 14(PP) — 16(PP_0) — 17(PP_1) — 18(PP_2) — 14(PP) — 20(RRC_1)—21(RRC_2)—13(NP_2)—1(S_0)—2(S_1)—6(VP_0) — 8(VP_2) — 9(VP_3) — 2(S_1) — 3(S_2) — success. The following figure shows the detailed procedures.

D. Contrasting the Complexity of the Transition States in Line with Varied Patterns

As Figure 21 shows, there are two backtrackings, four levels of pushing down and five hierarchical layers in the decoding process of Sentence 14. The decoding of Sentence 14 is clearly much more complicated than that of Sentence 13. The following table is the contrastive data concerning the 3 kinds of decoding procedures of Sentence 13 and Sentence 14.

As shown in Table VI, the hierarchical levels of the patterns of "Vtr+NP + PP", "Vtr+NP" and the GP sentences are respectively three, four and five, and the first two patterns have no backtrackings. Among the three patterns, the levels of pushing downs of GP sentences amount to as high as four levels, showing a much higher complexity than the former two patterns.

VIII. Conclusion

Globally ambiguous sentences and GP sentences have different syntactic structures. The ambiguous sentences entail a number of decoding selections and the successful decoding of GP sentences must exert a kind of single-selection system of semantic trigger, showing a higher cognitive burden. The reason why the decoding of GP sentences is more complex than the others lies in that the data structure of the GP sentences turns out to be a cognitive tree diagram with an indispensable understanding repetition. The augmented transition networks (ATNs) allow greater application of showing various states of decoding, thus significantly contributing to our understanding of the difference between globally ambigous sentences and GP sentences.

TABLE VI.
THE CONTRAST OF THE COMPLEXITY OF THE 3 KINDS OF DECODING PATTERNS OF SENTENCES 13 AND 14

Model		"Vtr+NP+PP" Model	"Vtr+NP" Model	Garden Path Sentence Model
Trace		Start-1-11-12-13-1-2-4-5-11-12-13-5-7-16-17-18-7-9-2-3-Success	Start-1-11-12-13-1-2-4-5-11-12-13-14-16-17-18-14-5-8-9-2-3-Success	Start-1-11-12-13-1-2-4-5-11-12-13-5-7-16-17-18-7-9-2-3-backtracking Restart-1-11-12-13-19-20-11-12-13 20-21-15/13-1-2-backtracking 14-16-17-18-14-20-21-15/13-1-2-6-8-9-2-3-Success
Hierarchy		3	4	5
Backtracking		0	0	2
Push	1-level	2	2	2(backtracking)+1(backtracking)+2
	2-level	2	1	2(backtracking)+1(backtracking)+1
	3-level	0	1	1(backtracking)+1
	4-level	0	0	1
	Total Push	4	4	7(backtracking)+5
Pop	1-level	2	2	2(backtracking)+1(backtracking)+2
	2-level	2	1	2(backtracking)+1(backtracking)+1
	3-level	0	1	1(backtracking)+1
	4-level	0	0	1
	Total Pop	4	4	7(backtracking)+5
Push and Pop Times		8	8	14(backtracking)+10
Complexibility		★	★★	★★★

Acknowledgment

This research is supported in part by grants of YB115-29 and YB115-41 from "the Eleventh Five-year" research projects of Chinese language application, and grant 08BYY046 from the National Social Science Foundation.

References

[1] T. G.Bever, "The cognitive basis for linguistic structures," In J. R. Hayes, Ed. *Cognition and the Development of Language*, New York: John Wiley and Sons, 1987, pp. 279-352.

[2] B. L.Pritchett, "Garden path phenomena and the grammatical basis of language processing," *Language*, 1988, pp. 539-576.

[3] Y. H. Jin, "Semantic analysis of Chinese garden-path

sentences," in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, 2006.

[4] K.Christianson, et al, "Thematic roles assigned along the garden path linger," *Cognitive Psychology*, 2001, (42), pp. 368–407.

[5] D. J. Foss and C. M. Jenkins, "Some effects of context on the comprehension of ambiguous sentences," *Journal of Verbal Learning and Verbal Behavior*, 1973, (12), pp. 577.

[6] K. G. D. Bailey and F. Ferreira, "Disfluencies affect the parsing of garden-path sentences," *Journal of Memory and Language*, 2003, (49): pp.183–200.

[7] Zh. W. Feng, "The automatic arithmetic of Garden Path Sentences," *Contemporary Linguistics*, 2003, (4), pp 339-349.

[8] B.Roark, "Robust garden path parsing," *Natural Language Engineering*, 2004, (10), pp. 1 – 24.

[9] P. F.Yu and J. L.Du, "Automatic analysis of textual garden path phenomenon: A computational perspective," *Journal of Communication and Computer*, 2008, (10), pp. 58-65.

[10] J. L. Du and P. F. Yu, "Towards the processing breakdown of syntactic garden path phenomenon: A semantic perspective of natural language expert system," *Journal of Communication and Computer*, 2008, 5(11), pp. 53-61.

[11] W. A. Woods, "Transition network grammars for natural language analysis," *CACM*, 1970, (13), pp. 591-606.

[12] R.M. Kaplan, "Augmented transition networks as psychological models of sentence comprehension," *Artificial Intelligence*, 1972, (03), pp. 77-100.

[13] G. Kempen, "Computational models of syntactic processing in human language comprehension," In: T. Dijkstra and K. De Smedt, Eds., *Computational Psycholinguistics: Symbolic and Subsymbolic Models of Language Processing*, London: Taylor & Francis, 1996, pp.192-220.

[14] N. D. Maxfield, J. M. Lyon, and E. R. Silliman, "Disfluencies along the garden path: Brain electrophysiological evidence of disrupted sentence processing," *Brain and Language*, 2009, 111(2), pp. 86-100.

[15] M. P. Wilson and S. M. Garnsey, "Making simple sentences hard: Verb bias effects in simple direct object sentences," *Journal of Memory and Language*, 2009, 60(3), pp. 368-392.

[16] Y. Choi and J. C. Trueswell, "Children's inability to recover from garden paths in a verb-final language: Evidence for developing control in sentence processing," *Journal of Experimental Child Psychology*, 2010, 106(1), pp. 41-61.

[17] K. R.Christensen, "Syntactic reconstruction and reanalysis, semantic dead ends, and prefrontal cortex," *Brain and Cognition*, 2010, 73(1), pp. 41-50.

[18] K. Christianson, S. G. Luke, and F. Ferreira, "Effects of plausibility on structural priming," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 2010, 36(2), pp. 538-544.

[19] M. Bader and J. Haussler, "Resolving number ambiguities during language comprehension," *Journal of Memory and Language*, 2009, 61 (03), pp. 352-373.

[20] W. Teubert, *Meaning, Discourse and Society*. Cambridge: United Kingdom, 2010.

[21] N. D.Patson, et al, "Lingering misinterpretations in garden-path sentences: Evidence from a paraphrasing task," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 2009, 35(1), pp. 280-285.

[22] J. L. Du and P. F. Yu, "Towards an algorithm-based intelligent tutoring system: Computing methods in syntactic management of garden path phenomenon," *IEEE International Conference on Intelligent Computing and Intelligent System*, 2010, pp. 521-525.

[23] E.Malaia, R. B. Wilbur, and C. Weber-Fox, "ERP evidence for telicity effects on syntactic processing in garden-path sentences," *Brain and Language*, 2009, 108(3), pp. 145-158.

[24] N. D. Patson and F. Ferreira, "Conceptual plural information is used to guide early parsing decisions: Evidence from garden-path sentences with reciprocal verbs," *Journal of Memory and Language*, 2009, 60(4), pp. 464-486.

[25] T. F.Jaeger, "Redundancy and reduction: Speakers manage syntactic information density," *Cognitive Psychology*, 2010, 61(1), pp. 23-62.

[26] A.Staub, "Eye movements and processing difficulty in object relative clauses," *Cognition*, 2010, 116(1), pp. 71-86.

A Systematic Approach to Context Aware Service Design

Tao Lu
School of Management Science and Engineering, Dalian University of Technology, Dalian, China
Email: lutao@dlut.edu.cn

Jie Bao
School of Management Science and Engineering, Dalian University of Technology, Dalian, China
Email: baojie@mail.dlut.edu.cn

***Abstract*—Timely design, development and marketing of new service with creative and innovative features are essential for service firm. Context-aware service, in which contextual information is used to adapt the behavior of service automatically to user's situation and need, is recognized to open up new opportunities for the service creation and service innovation. However, despite of technological possibility, context-aware service is still in its infancy. Lack of systematic methodology for service logic design is one of problems limiting the usage of contextual information to greatest extent. In this paper, we propose petri net as formalism for building a logic model of context-aware service. We make a further study about the dynamic behaviors and properties of the context aware service model, such as deadlock, invalid transition and inconsistency, etc. On this basis, we present a scenario-based method to design context-aware service model. A guidance system for elderly with dementia is designed by this method and the result has partly proved the effectiveness of our model and algorithm.**

***Index Terms*—context-aware, service design, scenario-based, petri net**

I. INTRODUCTION

Emerging systems of 3G and beyond make it possible to acquire contextual information from sensors, localization systems, RFID tags and other mobile devices[1]. The information can be used to adapt the behavior of services automatically to the user's situation and need, resulting in so-called context-aware services[2]. Among them, location-based service, such as in-car navigation system which has been available and successful for many years, is an important type of context-aware application. Context-aware service is very similar to location-aware service, except that a wide variety of contextual information is utilized. For example, mobile tourist guide system recommends place to visitor based on the current location, personal preference and visitor distribution. Pervasive healthcare system monitors patient's vital sign of life (e.g. blood pressure, heartbeat and temperature) through sensors anytime and anywhere. When any of these vital signs become abnormal, alerts are issued and certain reactions are scheduled. Context awareness may increase the usability of the service, and adds value. It has been recognized that context-aware service opens up new opportunities for the service creation and service innovation.

Context-aware service environment is generally complicate and dynamic. Unlike static desktop computer-based system that is driven by, or heavily relies on user's explicit input as command, context-aware systems stress delivering services to users in natural and non-intrusive way, without distracting them from their tasks[3]. This requires the underlying service logic of context-aware system should be highly autonomous. Context, as implicit input to system, should be utilized as fully as possible to reduce the interaction between user and system, and personalize service. Therefore it is unsuitable to simply copy service scenario of desktop computer-based system. Context-aware applications need to draw on the cognitive science and user experience in the design process. Only on this basis building context-aware service that make it a benefit, rather than a hindrance to users [3].

Most of current design frameworks for context-aware application are predominantly software-oriented[3], in which service logic design process is ignored under the assumption that the service logic is explicit. Service logic formulation process is actually sensor-driven, and the context-awareness is generally additional function to the service. This is very practical and reasonable in early stage of the technology development, when the available contextual information is limited. However, with more and more information sources providing different types of context which can be used by the service, service need to be designed in high-level and from holistic perspective.

In this paper, we propose a systematic approach to context-aware service design. Our work is based on the assumption that there exists appropriate technology and methodology to implement the service logic once it has been established, which is the area studied extensively by middleware design and software engineering.

The rest of the paper is organized as follows. Section 2 presents the related works in systematic methodology of service development. Section 3 proposes a unified logic model of context-aware application. We also explore the service logic design method based on scenario analysis in

doi:10.4304/jcp.7.1.207-217

section 4, and a design case is discussed in section 5. Section 6 concludes the paper.

II. Related Work

According to the widely accepted formal definition of context by Dey and Abowd[4], "context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves."

Systematic methodology of service development is researched in service engineering field, where service engineering is specialization of software engineering that targets the development of applications for consumption by end-users[5]. Here the concept of service is narrower, referring to a set of software which has certain functions. Among various methodologies, model-driven architecture (MDA) is regarded as more suitable [5]. Achilleos et al. propose a model-driven development process based on MDA paradigm. Designers can define context model at an abstract level and the definition of proactive rules for modeling context-aware behavior by a dynamic petri net is introduced[6,7]. Serral et al. propose a method applying the MDA guidelines for developers to properly specify context information in a set of models and to automatically generate the code of context-aware pervasive systems from these models[8]. Kapitsaki et al propose architecture and a model-driven methodology for development of context-aware composite applications[9].

In general, these efforts focus on implementing service logic efficiently and flexibly. The methodology we propose in this paper can be viewed as complementary to the work above. The main point of the methodology is to transform service scenarios into effective service logic model, which is independent of sensor and platform. In our early research, we have proposed a context aware service petri net model which focuses on the structural aspect [10]. In this paper we also consider the dynamic properties of service and reflect these properties in the presented model.

III. A Unified Framework of Context-awareness

Context-awareness implies determining the situation of the user and environment, and tailoring operation to the current context in user-friendly and effective way. We present a formal model in logical level, which can cover the wide variety of possible applications.

A. Context model and Context Reason

Let Θ be the set of concept representing the semantic meaning of context. We represent context as first-order logic predicate, which is often adopted in context model[11]. Formally, our model has basic form of $Predicate(a_1,a_2,...a_k)$, Where $Predicate \in \Theta, a_1,a_2,...a_k$ are the arguments of the predicate.

Since what is represented in service logic model is the relation of context and corresponding action being triggered, context in model is actually context type description, or condition that the context satisfies. So by using variable and set description in argument list to represent context type and conditions. For example, *network_connection_device*(*x*,{*PDA,laptop,cell-phone*}), or simply, *network_connection_device*(*x*, *portable*), refers to the context that the user access network with portable device, including PDA, laptop, and cell-phone. Context representation in following part of this paper refers to context type representation, which we will not mention specifically.

Contexts can be classified into two categories based on the means by which context is obtained, i.e. sensed context and inferred context. Sensed context is acquired from physical sensor or virtual sensor. For instance, physical location can be got through GPS, gender and age can be acquired from customer database which can be regarded as virtual sensor here. Sensed context reflects the basic facts that can be detected.

More facts about the situation can be inferred on the basis of sensed context and domain knowledge by context reasoning. This kind of context is called inferred context. For example, from the sensed context that there are 10 people in Room 203 and powerpoint is running at the same time, it can be deduced that there is a presentation in Room 203.

Generally, knowledge in the form of rule can be expressed as follows:

$$\bigwedge_{i=1}^{k} context_i(x_{i_1},x_{i_2},...x_{i_m}) \rightarrow context(y_1,y_2...y_n)$$

Where $context_i, i=1,2,...k,$ and $context$ are predicates.

Definition 1. Let S be set of sensed context. The inferred context set I_K is defined recursively as follow:

(1) If $\Phi \subseteq S$, $\Phi \rightarrow_K \alpha$ then $\alpha \in I_K$

(2) If $\Phi \subseteq S \cup I_K$, $\Phi \rightarrow_K \alpha$ then $\alpha \in I_K$

Where $\rightarrow_K$ denotes the entailment operation by the knowledge base *K*. ■

Knowledge base here should be interpreted in broader sense, including explicit knowledge, as well as implicit knowledge and corresponding process. According to this definition, if "powerpoint is running in Room 203" is a sensed context, through knowledge base, we can infer there is a presentation in Room 203, so the context "Presentation in Room 203" is an inferred context. If we know another sensed context "More than five people sit in chair", then we can know there is a conference in Room 203. Hence, "Conference in Room 203" is also an inferred context.

We define a special context, true context, denoted as c_{true}, which is always true. It can be viewed as a type of special sensed context.

We use $action\_result(\Phi,a)$ to represent context resulted from taking action *a* in context Φ.

Let *A* be a set of action taken by the service.

Definition 2. Let R_A be a set of result context by action set *A*. R_A is defined recursively as follow:

(1) If $\Phi \subseteq S \cup I_K$, $a \in A$, $\beta = action\_result(\Phi, a)$, then $\beta \in R_A$

(2) If $\Phi \subseteq S \cup I_K \cup R_A$, $a \in A$, $\beta = action\_result(\Phi, a)$, then $\beta \in R_A$ ■

From the definition, we can see that the result context is not just determined by the action, but also the context within which the action is performed. Result context can also be regarded as precondition to trigger another action. Hence, the result context actually represents process information, that the contexts having occurred and the actions having been performed.

B. Context-aware service model

Context awareness exploits contextual information to adapt the service's behavior. Context-aware service model is the logic model which represents the service reaction in certain contexts.

(i). Definition of Context-aware service model

We employ petri net, which is regarded as proper to model dynamic system, to model context aware service. In our context-aware service model, a context is represented by a place and an action is represented by transition. The flow relation reflects the condition and result relation between context and action. With the aim of adaption to the characteristic of context-awareness, we put some restrictions on the model. First, The place representing sensed context or inferred context have a token when it is true, have no token when it is false, and does not transfer token when the transition in its postcondition set is fired. In other word, the acquiring and losing of tokens in these places depends on its value of corresponding context, having no relation with transition. Therefore the indegree of every place representing sensed context and inferred context is zero. Second, we define that every place can be used only one time to trigger the transition in its postcondition set. At last, the model must be acyclic digraph.

Definition 3. We define context-aware service petri net as follow:

(1) Context-aware service petri net is a tuple $CSPT = (P, T, F, f_P, f_T)$, where P is a set of places, T is a set of transitions, F is a set of arcs, which $F \subseteq (P \times T) \cup (T \times P)$, $f_P : P \to S \cup I_K \cup R_A$, which is function assigning context to each place, $f_T : T \to A$, which is function assigning action to each transition. The precondition and postcondition of places and transactions is denoted as follow:

$\forall x \in P \cup T,\ \cdot x = \{y \mid (y, x) \in F\}$;

$\forall x \in P \cup T,\ x \cdot = \{y \mid (x, y) \in F\}$.

(2) Service petri net $CSPT = (P, T, F, f_P, f_T)$ should satisfy following rules.

Rule 1: $\forall p \in P$, if $f_P(p) \in S \cup I_K$, then $d^-(p) = 0$, if $f_P(p) \in R_A$, then $d^-(p) = 1$, where $d^-(p)$ is indegree of p.

Rule 2: $\forall t \in T\ \ t\cdot = \{\beta \mid f_P(\beta) = action\_result(\cdot t, t)\}$, i.e. the outplaces of the action transition are its result context places.

Rule 3: (P, T, F) is an acyclic digragh. ■

Definition 4. Given $CSPT = (P, T, F, f_P, f_T)$, we define a state of it as a pair (M, δ), where M is a marking, $M: P \to \{0,1\}$, and $\delta: P \to \{0,1\}$, which is function indicating the times of places having been used to fire transition. ■

We use (M_0, δ_0) to represent initial state of service petri net, where M_0 and δ_0 are defined as follows: for every places p in P, if $f_P(p) = c_{true}$, then $M_0(p) = 1$, otherwise $M_0(p) = 0$; for every places p in P, $\delta_0(p) = 0$.

Definition 5. Given $CSPT = (P, T, F, f_P, f_T)$ and a state (M, δ), a transition t is said to be enabled, if and only if: $\forall p \in \cdot t, M(p) = 1 \wedge \delta(p) = 0$. ■

Definition 6. Given $CSPT = (P, T, F, f_P, f_T)$, a state (M, δ) and an enabled transition t, on firing t, the state (M, δ) is changed to (M', δ'), such that:

(1) if $p \in t \cdot$ then $M'(p) = 1$

(2) if $p \in \cdot t \wedge f_P(P) \in R_A$ then $M'(p) = 0$

(3) if $p \in \cdot t$, then $\delta'(p) = 1$, otherwise $\delta'(p) = \delta(p)$ ■

For example, after the firing of transition t_1, the state shown in figure 1-a is changed to state in figure 1-b.

Because the tokens in sensed or inferred context places depend on the value of the context, there exist state changes not produced by firing transition. Therefore two states only different in number of tokens in sensed or inferred context places can be viewed as equal states, such as state in figure 1-a and state in figure 1-c are equal states, which are only different in number of tokens in sensed or inferred context places.

Definition 7. Given $CSPT = (P, T, F, f_P, f_T)$ and two states (M_1, δ_1), (M_2, δ_2), the two states are equal, denoted as $(M_1, \delta_1) = (M_2, \delta_2)$, if and only if :

$\forall p \in P : (\delta_1(p) = \delta_2(p)) \wedge (f_p(p) \in R_A \to M_1(p) = M_2(p))$ ■

Given a transition t and two states (M_1, δ_1) and (M_2, δ_2), if t is enabled in one of equal state of (M_1, δ_1) including (M_1, δ_1) itself, it is denoted as $(M_1, \delta_1)[t$. If $(M_1, \delta_1)[t$ and t is fired, changing to state (M_2, δ_2), we denote this process as $(M_1, \delta_1)[t\rangle(M_2, \delta_2)$.

Suppose $t_m, t_{m+1}, ..., t_n$ is a transition sequence, $(M_{m-1}, \delta_{m-1}), (M_m, \delta_m), ...(M_n, \delta_n)$ is a state sequence, if $(M_{i-1}, \delta_{i-1})[t_i\rangle(M_i, \delta_i)$, $m \le i \le n$, then it is denoted as $(M_{m-1}, \delta_{m-1})[t_m t_{m+1} ... t_n\rangle(M_n, \delta_n)$

Given $CSPT = (P, T, F, f_P, f_T)$ with initial state (M_0, δ_0), state (M_n, δ_n) is a reachable state, if and only if there exists a transition sequence as $t_1, t_2, ...t_n$, such that $(M_0, \delta_0)[t_1 t_2 ... t_n\rangle (M_n, \delta_n)$

Definition 8. Context-aware service is a tuple $CS = (S, K, I_K, CSPT)$, where S is a set of sensed context, and K is knowledge base, I_K is a set of inferred context and $CSPT$ is a set of context-aware service petri net. ■

Therefore, according to the above model we proposed, context-aware service design including determining sensed context set, knowledge base, inferred context set and service petri nets. Result context set can be determined when designing service petri net.

(ii). Property of context-aware service model

According to the definition 5 and 6, we can deduce that each context place can only be used one time to fire the action transition. Thus common context places in precondition sets may result in mutual exclusive relation.

However as a precondition to invoke action, one context may be needed not only one time. But using one place to represent this context may result in deadlock in execution, as shown in figure 2-a. This problem can be solved by representing same context in two places, as figure 2-b. Result context can also be used like this, i.e. action can produce several result context places mapping to same context (i.e. with same meaning) to invoke actions.

Definition 9. Given $CSPT = (P, T, F, f_P, f_T)$, $t_1, t_2 \in T$, t_1, t_2 are mutually exclusive, if and only if $\cdot t_1 \cap \cdot t_2 \neq \varphi$. ■

In order to be easy for analyzing the performance of the model, we use true context as common context to realize mutually exclusion. For example, in figure 3-a, t_1, t_2 are mutually exclusive.

From the definition, we can easily deduce that the necessary condition of enabling a transition *t* is that all predecessor transitions of *t* have been fired. Therefore, for two exclusive transitions t_i and t_j, the successor transitions of t_j will not be enabled if t_i has been fired, and vice versa.

Improper exclusion between transitions may result in some transitions never being enabled in any reachable state. Figure 3-a shows an example, in which transition t_3 can't be enabled no matter which transition, t_1 or t_2, is fired.

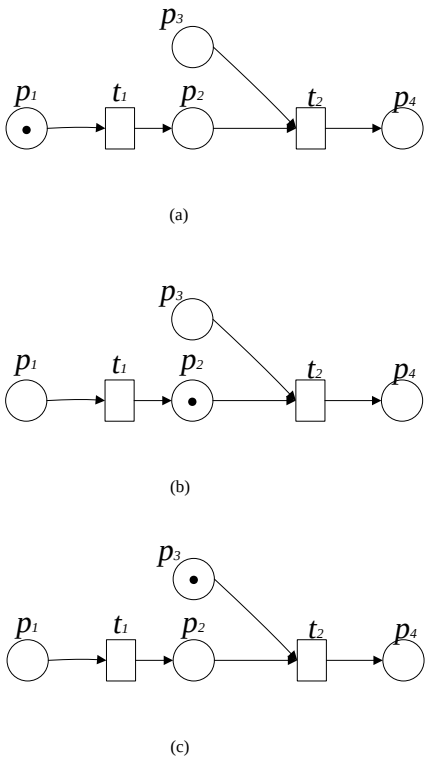


Figure 1. Some examples of the context-aware service model

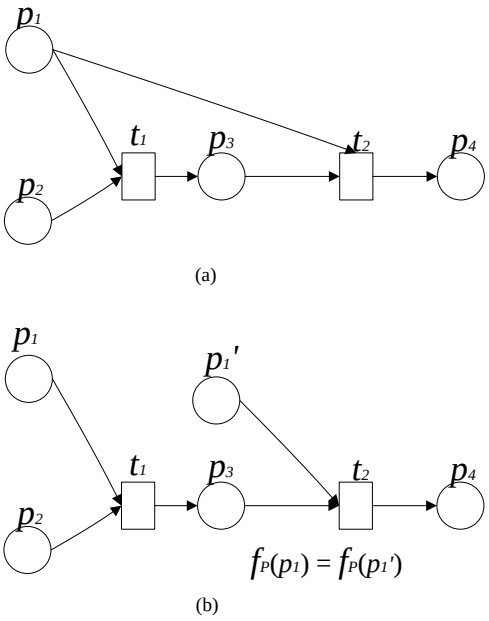


Figure 2. Deadlock caused by common place and solution

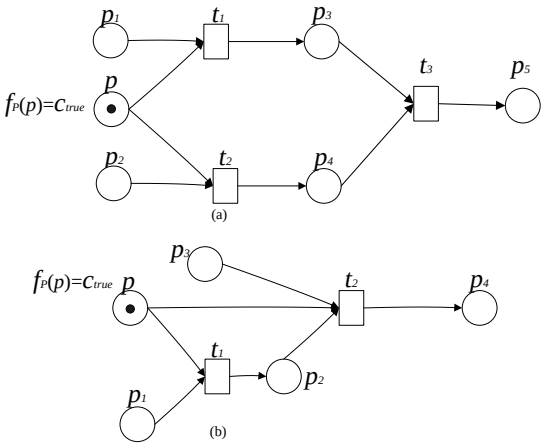


Figure 3. Exclusive transitions leading to transition never be fired

Definition 10. Given $CSPT = (P, T, F, f_P, f_T)$ with an initial state (M_0, δ_0), $t \in T$, we call t an invalid transition if and only if there does not exist a reachable state, such that t can't be enabled in this state or its equal states. ■

Proposition 1 Given $CSPT = (P, T, F, f_P, f_T)$, t is an invalid transition, if and only if t has a mutually exclusive predecessor transition or has at least two mutually exclusive predecessor transitions

Proof. sufficiency. If t has an mutually exclusive predecessor transition, then apparently the firing of predecessor transition disable firing of t, t can't be enabled if the predecessor transition have not been fired. If t has at least two mutual exclusive predecessor transitions, then in any time at least one of them can't be fired. Therefore t can't be enabled.

necessary. Since we assume that all contexts have possibility to be true, it suffices just to consider the result context. If t is an invalid transition, there is at least one result context place in the precondition set of t which will never get a token. Let N be subgraph which is the union of maximal directed path to t. Since N is acyclic digraph, there is $t_1, p_1, t_2, p_2, ...t_k, t$, which is topologic sequence of places and transitions in N. Therefore, we can prove that $(M_0, \delta_0)[t_1 t_2 ... t_k t$

According to this assumption, $(M_0, \delta_0)[t_1$, if t_1 is fired, $M(p_1) = 1$. Since t_1 has no exclusive transitions in $t_1, t_2, ...t_k$, firing t_1 will not change the tokens and times of firing places in precondition set of $t_2, ...$ and t_k. If $(M_0, \delta_0)[t_1 ... t_i > (M_i, \delta_i)$, then for $t_j \in \{t_{i+1}^{pre} \mid t_{i+1}^{pre} \cdot \subseteq \cdot t_{i+1}\}$, t_j has been fired. Hence $(M_i, \delta_i)[t_{i+1}$, i.e. t_{i+1} is enabled in a state equal to (M_i, δ_i). Therefore $(M_0, \delta_0)[t_1 t_2 ... t_k t$. This contradicts to the hypothesis that t is an invalid transition. ■

For example, we can easily find out in figure 3-a, t_3 has two mutual exclusive predecessor transitions, t_1 and t_2. No matter which transition is fired, t_3 can't be enabled.

Another example is shown in figure 3-b. t_2 has an mutually exclusive predecessor transition t_1. If t_1 is fired, context place p will lose the token, result in that t_2 can't be enabled. If t_1 has not been fired, t_2 can't be enabled because context place p_2 has no chance to acquire a token.

With the aim of not making the model too complex, we require that in our model, exclusion occurs in the selection of two branches without common successor, not in every pair of transitions which can't be fired concurrently. This ensures that the model having no invalid transition and not be too complex on the other hand.

Ⅳ. SERVICE DESIGN BASED ON SCENARIO.

Context-aware service logic describes the service process in various situations, in particular the service reaction to the context change. We propose a scenario-based method in logic designing process.

Scenarios are stories about people and activities[9]. Each scenario describes a partial behavior arising in restricted situation[12], and can be regarded as a particular case of how the system is to be used. Scenarios are recognized to be very appropriate for reactive systems external behavior description[12].

There are three stages in our proposed method: describing scenario in service petri net models, constructing knowledge base, and model refinement.

A. Scenario description in service petri net models

Scenario is a story describing what happens to the user and the service reaction. The original scenario is acquired by experience and imagination. Developers, designers or the potential users can give their scenarios from different viewpoints.

A scenario is composed of actors and events. Actors are replaced by variables in description. There are two types of event: internal event, which corresponds to an action taken by the system, and external event, which is the human activity or environment change.

Let E_{int} be a set of internal events, E_{ext} be a set of external events, E_{result} be a set of result event, for each event in which has a corresponding event in E_{int}.

Suppose $e' \in E_{result}$, its corresponding internal event is e, $e \in E_{\text{int}}$, we define the semantic meaning of result event e' is that e has happened. This is consistent with result context we have defined in last section.

An internal event which represents a system action may also results in other consequence, such as environmental change or human activity. This result is regarded as the external event, which can be sensed of inferred.

Therefore, we can describe scenario in service petri net model structure by analyzing the relation of events in semantic level. We use nodes to represent events. Let

$G=(P,T;F)$, where $P=E_C \cup E_R$, $T=E_{\text{int}}$. Every arc in F is determined as follows rules.

Rule 1: For $v_i,v_j \in P \cup T$, if $v_i \in E_{ext} \cup E_{result}$, $v_j \in E_{\text{int}}$, v_i is one of condition to invoke v_j, $d^+(v_i)=0$, then $F=F\cup\{(v_i,v_j)\}$

Rule 2: For $v_i,v_j \in P \cup T$, if $v_i \in E_{ext} \cup E_{result}$, $v_j \in E_{\text{int}}$, v_i is one of condition to invoke v_j, $d^+(v_i)>0$, then insert an node v_i',i.e. $P=P\cup\{v_i'\}$ such that v_i' represents the same external event as v_i, $F=F\cup\{(v_i',v_j)\}$

Rule 3: For $v_i,v_j \in P \cup T$, if $v_i \in E_{\text{int}}$, $v_j \in E_{result}$, v_j is result event of v_i, which means v_i has happened, then $F=F\cup\{(v_i,v_j)\}$

For example, we have a scenario: Bob's caregiver gives a medicine schedule. When it is time to take medicine, the system checks that Bob has not got his medicine from medicine cabinet, the system sends a remind information to Bob. Bob fetched medicine, the system checks and determines it is right type and amount, so finishes the service. The above scenario can be represented in petri net model, as shown in figure 4.

Apparently in the initial model developed from single scenario, which has no different options of every episode, there are no exclusive transitions.

B. Constructing context structure and knowledge base

In the initial petri net models, external event is regarded as place representing context. We use context description and parameter list to replace each event in E_{ext}. Then we analyze each context description and define its corresponding context type. Context types of sensed context can be directly determined by sensor data. When we define this kind of context type, we can just specify its semantic meaning and acquisition method.

By the context that we can't acquire directly, i.e. inferred context, we should define how to determine it from sensed context and knowledge. This can either be done by specifying the inferred context and its relationship with other contexts, or making full use of machine learning techniques to acquire the knowledge automatically.

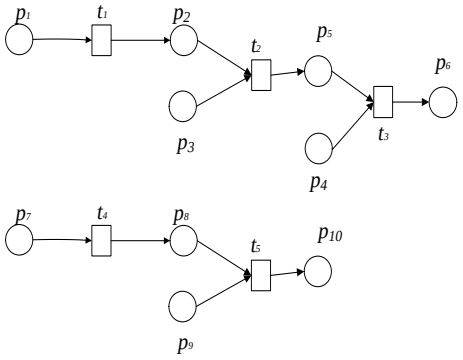
For example, the context "party" in one room can't be directly sensed. But if the sound, the light and the number of people in the room are sensed by the system, whether there is a party in the room can be deduced. Therefore, the corresponding knowledge can be expressed by a rule:

$$sound\_in\_room(x,>\beta)\wedge light\_in\_room(x,"=on")$$
$$\wedge number\_of\_people(x,>\gamma)$$
$$\Rightarrow activity\_in\_room(x,"party")$$

Where β,γ are predefined values.

Not all the knowledge can be represented explicitly in rules. The knowledge base is a broader concept here. For example, the high-level context like "lying down" can be learned from the individual audio and video data streams, such as in [13]. Actually the result model after training is knowledge.

Therefore, in design phase we need to specify the context relationship and the structure of knowledge base. However, some parameter and knowledge content can't be determined in advance.



p_1: scheduled time
p_3: having not taken medicine
p_4: having taken medicine
p_7: fetch medicine
p_9: right medicine
t_1: check if having taken medicine
t_2: remind to take medicine
t_5: permitting to fetch medicine
t_3: end of the service
t_4: check if it is right time and right type and amount

Figure 4. Petri net representation of scenario

It is possible that some contexts can't be determined by current technology and knowledge. The service model is ineffective if there is undetermined context. In this step, ineffective models should be deleted.

C. Model refinement

Initial models needing to be refined arise from two facts. First, the initial consideration may be imprecise or incompatible, especially in dealing with single scenario. Second, there may be same episode in similar scenarios, the corresponding model need to be synthesized.

There are two steps in model refinement: checking model consistency and model synthesis.

(i). Checking model consistency

Let $G_P: P(P)\to P(S\cup I_K)$, $G_P(X)=\{f_P(x)\,|\,x\in X\}$, where $P(P)$ and $P(S\cup I_K)$ are the power sets of places P and $S\cup I_K$ respectively.

If $G_p(\cdot t_1)/\{c^{true}\}\supset G_p(\cdot t_2)/\{c^{true}\}$ and $f_T(t_1)=f_T(t_2)$, then we can know that either $\cdot t_1$ or $\cdot t_2$ is imprecise. $G_p(\cdot t_1)/\{c^{true}\}\supset G_p(\cdot t_2)/\{c^{true}\}$ means that not considering true context, the set of contexts which the places in $\cdot t_2$ represent is the subset of contexts which the places in $\cdot t_1$ represent. If contexts represented by the places in $\cdot t_2$ are sufficient to trigger the action, then there is at least one context place in $\cdot t_1$ is redundant. On the contrary, if each context place in $\cdot t_1$ is necessary, then $\cdot t_2$ is not sufficient

to trigger the action. So developer should check the condition and the action to correct imprecise representation in such kind of cases.

Besides imprecise representation, there may be conflict caused by incompatible action. In a consistent model, if there are more than two transitions fired in same contexts, the action that the transitions represent should be compatible, or, in other word, the conflict action is not permitted.

Let $t_1,t_2 \in T$, if $G_p(\cdot t_1)/\{c^{true}\}=G_p(\cdot t_2)/\{c^{true}\}$, $f_T(t_1)$, $f_T(t_2)$ should be compatible. That is to say, if the contexts represented by the places firing t_1 and t_2 are same, the actions that t_1 and t_2 represent should be compatible.

More generally, if $\underset{p\in t_1}{\wedge} f_P(p) \rightarrow \underset{p\in t_2}{\wedge} f_P(p)$, $f_T(t_1)$ and $f_T(t_2)$ should be compatible.

$\underset{p\in t_1}{\wedge} f_P(p) \rightarrow \underset{p\in t_2}{\wedge} f_P(p)$ means the contexts enabling transition t_1 are true implies contexts enabling transition t_2 are true, For instance, if $G_P(\cdot t_1)=\{temperature(y,>30^\circ C)\}$, $G_P(\cdot t_2)=\{temperature(y,>25^\circ C)\}$, then we can easily conclude that $temperature(y,>30^\circ C) \rightarrow temperature(y,25^\circ C)$. Developer needs to guarantee that concurrently firing the two actions would not lead to confliction.

(ii). Model synthesis

Two transitions are viewed equivalent if they represent same action and occur in same context.

Definition 11. Given $t_1, t_2 \in T$, t_1 and t_2 are equivalent transitions, denoted as $t_1 \cong t_2$, if and only if $f_T(t_1)= f_T(t_2)$ and $G_p(\cdot t_1)/\{c^{true}\}=G_p(\cdot t_2)/\{c^{true}\}$ ■

Suppose $CSPT=(P,T,F,f_P,f_T)$,and $N_1=(P_1,T_1,F_1)$ $N_2=(P_2,T_2,F_2)$, N_1 and N_2 are the two connected components in $CSPT$. Let $T_{1,2}=\{t_1 \mid t_1 \in T_1$ and $\exists t_2 \in T_2$ such that $t_1 \cong t_2\}$, $T_{1,2}$ is the transition set of T_1 which has equivalent transition in T_2 . If $T_{1,2} \neq \phi$, then the two components need to be synthesized.

Similarly we can define $T_{2,1}=\{t_2 \mid t_2 \in T_2$ and $\exists t_1 \in T_1$ such that $t_1 \cong t_2\}$. Obviously, there exist a bijection $\varphi: T_{1,2} \rightarrow T_{2,1}$, such that $\forall t \in T_{1,2}, t \cong \varphi(t)$

Let $SuccT(t)=\{t' \mid \exists p \in P,$ such that $(t,p) \in F \wedge (p,t') \in F\}$ be a direct successor transition set of *t*. We perform synthesis by algorithm SynthesizeComponent shown in figure 5.

The synthesized model is a component which can also be synthesized with other component having equivalent transitions. Thus repeating the algorithm, we will get a refined model.

Proposition 2. Algorithm SynthesizeComponent will not produce invalid transitions in the synthesized model.

Proof. By Proposition 1, invalid transition is resulted from exclusion between transitions. Since the initial petri net model is developed from single scenario which has no branches and exclusions, exclusion occurs in neighbor transitions when synthesizing two components by algorithm. So for any two exclusive transitions, neither of them is successor of another, and they don't have common successor transitions when synthesized because they belong to different components. On the other hand, since the exclusive neighbor transitions are not equivalent transitions, successors of any transition will not have equivalent transition in successors of another. Hence the successors of two exclusive transitions will never be synthesized and the two exclusive transitions will not have common successor. According to proposition 1, no invalid transition will be produced in the model.

V. CASE STUDY

We adopt scenario-based method to perform logic design of a guidance system which helps elderly with dementia in their daily life. Impaired episodic memory, that is, memory of personally experienced events, is the symptom most commonly associated with mild to moderate dementia[14][15]. Older people with dementia often forget to do or doing something they have planed, such as taking medicine, visiting old friend, etc. Guidance system will remind them and help them to complete their activities. This will reduce the burden of Reminding, alerting and notifying caregiver (or doctor, etc.) are basic service of service is by no means simply providing reminding information as schedule. For example, elderly with dementia often forget to take medicine, and on the other hand if they have taken medicine, they may also forget they have done. Reminding without judge may lead to taking medicine second time which is not less harmful as not taking. Therefore the system should monitor the activities of elderly and provide proper help. In addition, providing minimal prompts is a one of criteria in designing such kind of system. This is to guarantee the elderly with dementia try their best to exercise their brains to delay the deterioration of dementia[16]. So the system should try to remind when the elderly really forget.

Assume that elderly, beds, medicine cabinet and other items are equipped with wireless sensors and radio frequency identity (RFID) tags, enabling that the activity such as sleeping, getting medicine or putting the medicine back can be detected by the system. We develop different scenarios from various views, and each

Algorithm SynthesizeComponent

Input $N_1 = (P_1, T_1, F_1)$, $N_2 = (P_2, T_2, F_2)$

Output $N_{1\cup 2} = (P_{1\cup 2}, T_{1\cup 2}, F_{1\cup 2})$ which synthesizing N_1 and N_2

Begin /* begin 0 */

$P_{1\cup 2} = P_1 \cup P_2$, $T_{1\cup 2} = T_1 \cup T_2$, $F_{1\cup 2} = F_1 \cup F_2$

for each t in $T_{1,2}$ and $SuccT(t) \not\subset T_{1,2}$

for each t^{Succ} in $SuccT(t)$ and $t^{Succ} \notin T_{1,2}$

begin /* begin 1*/

$P^{result} = \cdot t^{succ} \cap t \cdot$

$F_{1\cup 2} = F_{1\cup 2} \cup \{(\varphi(t), p) \,|\, p \in P^{result}\}$

$F_{1\cup 2} = F_{1\cup 2} - \{(t, p) \,|\, p \in P^{result}\}$

for each $t^{neighbor}$ in $SuccT(\varphi(t))$

if $t^{neighbor}$ and t^{succ} are exclusive

begin /*begin 2*/

CreatePlace(p)

$f_P(p) = c^{true}$

$F_{1\cup 2} = F_{1\cup 2} \cup \{(p, t^{neighbor}), (p, t^{succ})\}$

end / * end of begin 2*/

end / * end of begin 1*/

$P_{1\cap 2} = \{p \,|\, \exists t \in T_{1,2}$ such that $p \in \cdot t \vee p \in t \cdot\}$

$F_{1\cup 2} = F_{1\cup 2} - \{(v_i, v_j) \,|\, v_i \in P_{1\cap 2} \cup T_{1,2}$ or $v_j \in P_{1\cap 2} \cup T_{1,2}\}$

$P_{1\cup 2} = P_{1\cup 2} - P_{1\cap 2}$

$T_{1\cup 2} = T_{1\cup 2} - T_{1,2}$

end / * end of begin 0*/

Figure 5. Algorithm of model synthesis

including the situations that elderly may encounter and corresponding reactions of the system. Each scenario is represented in initial petri net model, as shown in figure 6. When it's time for the elderly to take medicine and meanwhile the medicine in cabinet has not been fetched, system reminds the elderly in different forms according to the situation. Specifically, If the elderly is sleeping at that moment, then system deals with that information, else system reminds him/her to take medicine and scenario 1 and 2 describe these two situations. When the elderly moves towards the cabinet to fetch medicine, system responses to different conditions. Particularly, if it's right time for taking medicine, then the system records the behavior without sending out any information. If it is wrong time, the system sends out remind message asking him/her to put the medicine back, and monitors the successor activity to guarantee that the medicine are put back in time. In addition, some medicine, called emergent medicine in the case, e.g. pain reliever, are not in schedule list but are needed sometimes. When system checks these kinds of medicine are taken from cabinet, it reminds the elderly to read drug description and records the event to avoid taking the second time during a short period. These different conditions are shown in scenario 3, 4, 5 and 6. When the elderly puts the medicine back to the cabinet, the system checks whether the medicine is new or old, and responds to each situation. Scenario 7 and 8 describe this process.

In knowledge base construction phase, we just analyze every place in initial petri net model and determine the context type, without considering more details about acquiring every context, which heavily depends on the development framework. The contexts in the case are not complicate, and can be sensed or inferred.

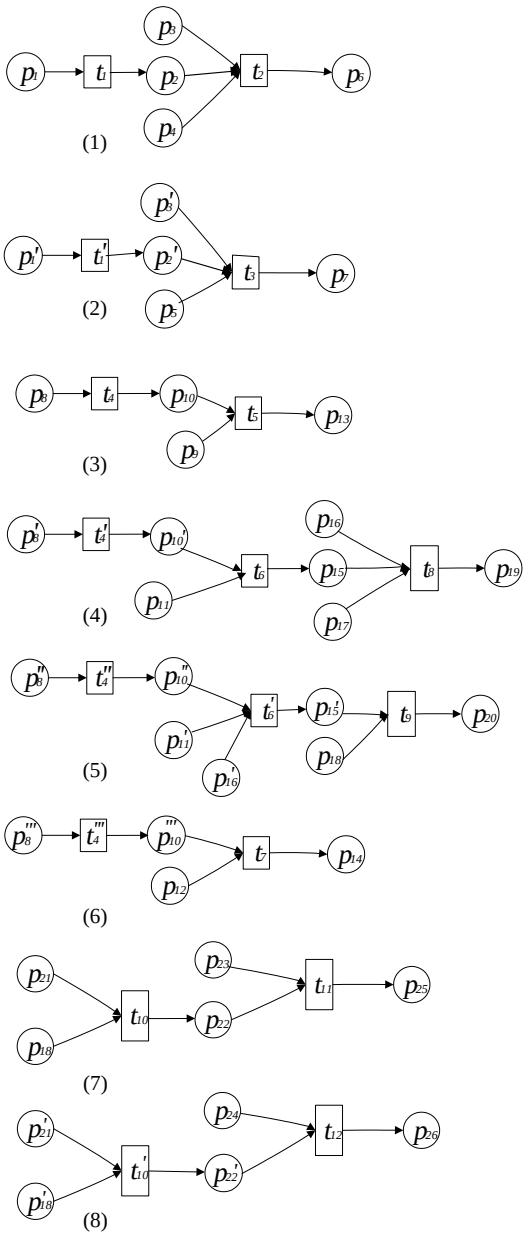
In model refinement phase, we check the consistency of the model. After checking the scenario 4 and 5 in figure 6, we can easily find out that $G_P(\cdot t_6') \supset G_P(\cdot t_6)$, $f_T(t_6) = f_T(t_6')$. Therefore, according to model consistency checking rule mentioned in section 4, we can know that there is a redundant context place. In our case, this redundant context place is p_{16}'. By checking our scenario we can see when the patient fetches the wrong medicine, no matter where he locates, the remind information should be send out immediately. Hence, the context *away_from_cabinet*(*x*) is redundant and it needs to be eliminated. In addition, some parts in the case also need

to be synthesized. For example, in scenario 1 and 2, we can see $G_P(\cdot t_1)=G_P(\cdot t_1')$, $f_T(t_1)=f_T(t_1')$, therefore, we can facilitate synthesizing the two scenarios according to algorithm SynthesizeComponent, and other scenarios can be synthesized similarly.

Figure 7 shows part of the final synthesized model, related with reminding to take medicine when it is scheduled time, system checking when fetching medicine from cabinet and putting back medicine. The component in figure 7-a is the synthesized result by scenario 1 and 2. Component in figure 7-b is the synthesized result by scenario 3, 4, 5 and 6 and the last component in figure 7-c is the synthesized result by scenario 7 and 8.

Ⅵ. Conclusion.

Context-aware service logic describes the reaction of service in certain context or to certain context change. In requirement analysis and logic design phase, service scenarios can help designers to develop rich types of how the service maybe used in dynamic environment. In this paper, we propose scenario-based top-down method to create and refine context-aware service logic. The method can be viewed as complementary to current service engineering research, which enables developers to implement service logic in flexible and efficient ways.



$f_P(p_1)=f_P(p_1')$: *shedule_time*(x,y)

$f_P(p_3)=f_P(p_3')$: ¬*fetched_medicine*(x,y)

$f_P(p_4)$: *sleeping*(x)

$f_P(p_5)$: ¬*sleeping*(x)

$f_P(p_8)=f_P(p_8')=f_P(p_8'')=f_P(p_8''')$: *fetched_medicine*(x,y)

$f_P(p_9)$: *right_time_and_medicine*(x,y)

$f_P(p_{11})=f_P(p_{11}')$: *wrong_time_or_medicine*(x,y)

$f_P(p_{12})$: *emergent_medicine*(x,y)

$f_P(p_{16})=f_P(p_{16}')$: *away_from_cabinet*(x)

$f_P(p_{17})$: ¬*put_back*(x,y)

$f_P(p_{18})=f_P(p_{18}')$: *put_back*(x,y)

$f_P(p_{21})=f_P(p_{21}')$: *no_system_request*(x,y)

$f_P(p_{23})$: *old_medicine*(x,y)

$f_P(p_{24})$: *new_medicine*(x,y)

$f_T(t_1)=f_T(t_1')$: check if x has taken medicine y

$f_T(t_2)$: deal with remind information when x is asleep

$f_T(t_3)$: remind x to take medicine y

$f_T(t_4)=f_T(t_4')=f_T(t_4'')=f_T(t_4''')$: check if it is right time for x to fetch y

$f_T(t_5)$: record that x has taken y

$f_T(t_6)=f_T(t_6')$: remind x that he/she has fetched wrong medicine

$f_T(t_7)$: remind x to read drug description and record

$f_T(t_8)$: remind x to put back medicine one more time

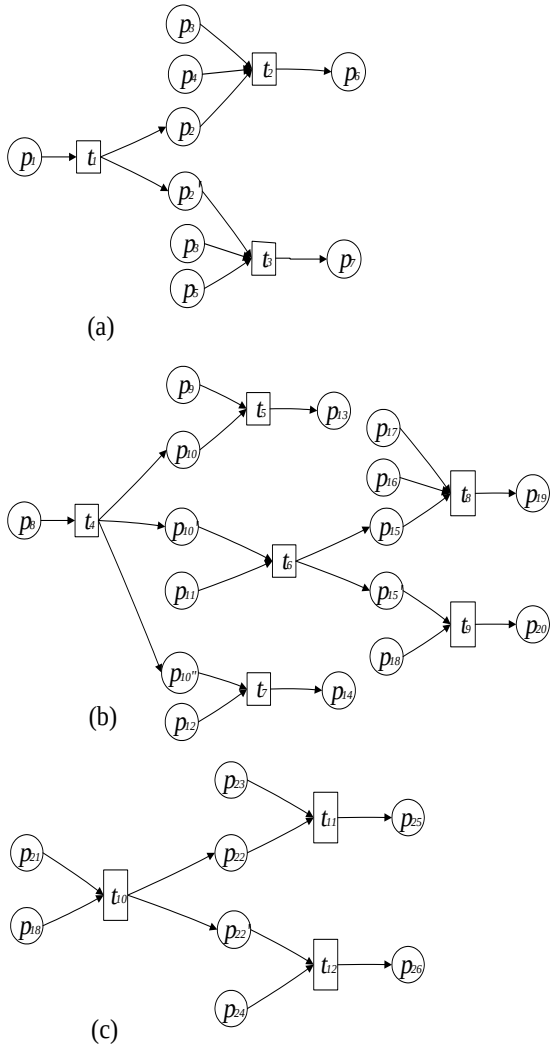
$f_T(t_9)$: end the service this time

$f_T(t_{10})=f_T(t_{10}')$: check the medicine and schedule

$f_T(t_{11})$: remind to take medicine as schedule

$f_T(t_{12})$: record medicine amount

Figure 6. Scenarios represented in initial petri net model service for elderly with dementia



$f_P(p_1)$: *shedule_time*(x,y)
$f_P(p_3)$: ¬*fetched_medicine*(x,y)
$f_P(p_4)$: *sleeping*(x)
$f_P(p_5)$: ¬*sleeping*(x)
$f_P(p_8)$: *fetched_medicine*(x,y)
$f_P(p_9)$: *right_time_and_medicine*(x,y)
$f_P(p_{11})$: *wrong_time_or_medicine*(x,y)
$f_P(p_{12})$: *emergent_medicine*(x,y)
$f_P(p_{16})$: *away_from_cabinet*(x)
$f_P(p_{17})$: ¬*put_back*(x,y)
$f_P(p_{18})$: *put_back*(x,y)
$f_P(p_{21})$: *no_system_request*(x,y)
$f_P(p_{23})$: *old_medicine*(x,y)
$f_P(p_{24})$: *new_medicine*(x,y)
$f_T(t_1)$: check if x has taken medicine y
$f_T(t_2)$: deal with remind information when x is asleep
$f_T(t_3)$: remind x to take medicine y
$f_T(t_4)$: check if it is right time for x to fetch y
$f_T(t_5)$: record that x has taken y
$f_T(t_6)$: remind x that he/she has fetched wrong medicine
$f_T(t_7)$: remind x to read drug description and record
$f_T(t_8)$: remind x to put back medicine one more time
$f_T(t_9)$: end the service this time
$f_T(t_{10})$: check the medicine and schedule
$f_T(t_{11})$: remind to take medicine as schedule
$f_T(t_{12})$: record medicine amount

Figure 7. Part of finally synthesized logic model service for elderly with dementia

By scenario-based method, developers and users can describe the target system from different point of views. Different scenarios can be systematically dealt with. The compatibility, consistency between scenarios can be checked semi-automatically and models can be synthesized automatically, where human activity is focused on semantic level. This will reduce the burden of developers.

However, the design process is also state-of-art. Experience and technology background play an important role in whole process. The semantic analysis can't be automatic. When designer give scenario description, he/she should select appropriate granularity of every event, which requires deeply understanding of service program.

In addition, we view every scenarios having same importance in this method. But not every scenario is the best way to provide the service and bad scenario will result in unreasonable logic model. We need a method to measure the scenario before we construct the model.

Our research work is just in preliminary stage. Further application of our method should be taken into account in future. The method need to be improved in order to be appropriate for more complicated applications.

ACKNOWLEDGEMENT

We would like to express our sincere appreciation to the anonymous reviewers for their insightful comments, which have greatly aided us in improving the quality of the paper.

This research work was supported by the National Natural Science Foundation of China (Grant No. 70771017, 70890080, 70890083)

REFERENCE

[1] Gabriella Castelli, Marco Mamei, Franco Zambonelli, "Engineering contextual knowledge for autonomic pervasive services," *Information and Software Technology*, vol.50, no.1-2, pp.36-50, 2008.

[2] Mark de Reuver, Timber Haaker, "Designing viable business models for context-aware mobile services," *Telematics and Informatics*, vol.2006, no.3, pp.240-248, 2009.
[3] Nicholas A. Bradley, Mark D. Dunlop, "Toward a multidisciplinary model of context to support context-aware computing," *Human–computer interaction*, vol.20, no.4, pp.403–446, 2005.
[4] Anind K. Dey, Gregory D. Abowd, Daniel Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human-Computer Interaction*, vol.16,no.2-4, pp.97-166, 2001.
[5] Georgia M. Kapitsaki, George N. Prezerakos, Nikolaos D. Tselikas, Iakovos S. Venieris, "Context-aware service engineering: A survey," *The Journal of Systems and Software*, vol.82, no.8, pp.1285–1297, 2009.
[6] Achilleas Achilleos, Kun Yang, Nektarios Georgalas, "Context modelling and a context-aware framework for pervasive service creation: A model-driven approach," *Pervasive and Mobile Computing*, vol.6, no.2, pp.281-296, 2010.
[7] Achilleas Achilleos, Kun Yang, Nektarios Georgalas, Manooch Azmoodech, "Pervasive service creation using a model driven petri net based approach," *International Wireless Communications and Mobile Computing Conference*, pp.309-314, Aug.6-Aug.8, 2008, Crete, Greece.
[8] Estefanía Serral, Pedro Valderas, Vicente Pelechano, " Towards the model driven development of context-aware pervasive systems," *Pervasive and Mobile Computing*, vol.6, no.2, pp.254-280, 2010.
[9] Georgia M. Kapitsaki, Dimitrios A. Kateros, George N. Prezerakos, Iakovos S. Venieris, "Model-driven development of composite context-aware web applications," *Information and Software Technology*, vol.51, no.8, pp.1244–1260, 2009.
[10] Tao Lu, Qian Hao, "Scenario-based context-aware service design," *7th International Conference on Service Systems and Service Management*, pp.509-514, Jun.28-Jun.30, 2010, Tokyo, Japan.
[11] Anand Ranganthan, Roy H. Campbell, "An Infrastructure for Context-Awareness Based on First Order Logic," *Pervasive Ubiquitous Computing*, vol.7, pp.353-364,2003
[12] J.M. Carroll, "Five Reasons for Scenario-Based Design," *Interacting with Computers*, vol.13, no.1, pp.43-60, 2000.
[13] R. Dssouli, S. Some, J. Vaucher, A. Salah, "A service creation environment based on Scenarios," *Information and Software Technology*, vol.41, no.11, pp.697–713, 1999.
[14] O. Brdiczka, J.L. Crowley, P. Reignier, "Learning situation models for providing context-aware services," *Proceedings of Universal Access in Human-Computer Interaction*, UAHCI 2007, in: Lecture Notes in Computer Science, vol. 4555, Springer, 2007.
[15] Hua Si, Seung Jin Kim, Nao Kawanishi, Hiroyuki Morikawa, "A Context-aware Reminding System for Daily Activities of Dementia Patients," *27th International Conference on Distributed Computing Systems Workshops*, June 22-June 29,2007, Toronto, Canada
[16] Joseph P. Wherton, "Andrew F. Monk. Technological opportunities for supporting people with dementia who are living at home," *International Journal of Human Computer Studies*, vol.66, no.8, pp.571–586, 2008.
[17] Kejun Du, Daqing Zhang, Xingshe Zhou, Mossaab Hariz, " Handling conflicts of context-aware reminding system in sensorised home," *Cluster Computing*, DOI 10.1007/s10586-009-0091-1.
[18] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, Daniele Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol.6, no.2, pp.159-256, 2010.
[19] Tarak Chaari, Dejene Ejigu, Frédérique Laforest, Vasile-Marian Scuturici, "A comprehensive approach to model and use context for adapting applications in pervasive environments," *The Journal of Systems and Software*, vol.80, no.12, pp.1973-1992, 2007.
[20] Chulhyun Kim, Suhwan Choe, Changwoo Choi, Yongtae Park, "A systematic approach to new mobile service creation," *Expert Systems with Applications*, vol.35, no.3, pp.762-771, 2008.
[21] Oh Byung Kwon, "Modeling and generating context-aware agent-based applications with amended colored Petri nets," *Expert Systems with Applications*, vol.27, no.4, pp.609–621, 2004.

A Self-adaptive and Real-time Panoramic Video Mosaicing System

Lin Zeng and Dexiang Deng
School of electronic information, Wuhan University, Wuhan, China
Email :{zenglin.whu, whuddx}@gmail.com

Xi Chen*
Institute of Microelectronics and Information Technology, Wuhan university, Wuhan, China
Email: robertcx@whu.edu.cn

Yunlu Zhang
School of Computer information, Wuhan University, Wuhan, China
Email: zhangyunlu850527@gmail.com

***Abstract*—In real-time monitoring process, resolution must be set against field vision. There are two problems in the real-time monitoring procedures, which are: the field vision is too small to capture the target completely and a larger field vision with low resolution . In order to solve these problems, a new type of real-time panoramic monitoring systems becomes necessary. In this paper, we proposed a new adaptive video mosaicing algorithm based on SIFT algorithm and template matching method. By real-time adjusting and updating the mosaicing position of adjacent two images to establish the optimal mosaic line, we can both made full use of the accuracy of feature matching and the simplicity of template matching. Experimental results show our algorithm not only has good matching rate but also reduces the time complexity, which make it very suitable for the application which requests high real-time performance.**

***Index Terms*—Video mosaicing, panoramic, SIFT, real-time**

I. INTRODUCTION

Traditional video monitoring system has limited filed vision, and there is a difficult problem on monitoring all events in 360 degree range at the same time, which all make the monitoring blind area, is necessary to the monitoring system improvement. The generation of panoramic video is one of the key methods. There are mainly three methods to generate panorama video based on three kinds of panoramic camera currently:

One kind is composed of single cameras equipped with fish eye lens. But fisheye lens has poor average resolution and irreversibility distortion on close range images. The inconsistency of viewpoint on various parts of image may lead to the difficulties of post process.

The second kind panoramic camera is to change the lights come into it by the principle of light refraction and light reflection, such as FullView [1] [2].It constructed a multilateral body with plane mirror, which make multiple camera view mirror overlap at one point virtually, this method effective constructs a panoramic and virtual camera that can capture the image video frame rate. Its filming image has high resolution, good effect and can achieve the video frame rate, makes it applicable to sports complex environment. With the help of adjusting exposure and brightness of the camera, it can form panorama video without post algorithm processing. It is the best way to establish real-time panoramic video mosaicing system, but the necessary auxiliary optical components require strict accuracy, the modeling procedures are quite tedious and time-consuming, at the meantime, the cost is high.

The finally kind panoramic camera is by the help of multiple cameras with different directions, for example, Flycam panoramic camera system [3]. This plan can capture different images on different angles and different spaces by multiple fixed position cameras and then to mosaic the scenes to get the panoramic images. It can also provide a higher resolution and just need ordinary camera to capture images. It can make any shape without any special equipment. But because of the inconsistent camera viewpoint, especially in the overlapping area of adjacent images existing long shot and close shot simultaneously, the ghost [5] it is hard to avoid.

Panoramic video mosaicing systems used for monitoring have the higher requirement of real-time and resolution, our system based on the third kind of panoramic camera motioned above, it is small in size and easy to installation and maintenance, and at the same time it can adjust the size of mosaicing area adaptively with high resolution by our Self-adaptive optimal mosaic line matching algorithm.

II. RELATED WORK

Panorama image generation mosaicing is one of the key methods to generate panorama image, by combining multiple overlapping image about the same scene into a panoramic images. A classic example panorama application is QuickTimer commercial software [6] by

*. Corresponding author: Xi Chen (Email: robertcx@whu.edu.cn)

doi:10.4304/jcp.7.1.218-225

Apple Company. It generates cylinder panorama image in some key point positions, and, on each key positions, it can realize the eye-gaze continuous variation and achieve scene roaming by the jump between key points positions.

Kyung and others advances an effective algorithm to build cylinder panoramic image mosaicing. By equidistance matching the images horizontal photographed by cameras, they generate the cylinder panoramic image based on horizontal motion model, and they can effectively Estimated the focal length of used cameras by dichotomy algorithm. It is the most commonly used methods currently. And our paper uses it as a foundation. Shum and Szeliksi [7] put forward a panoramic image mosaicing algorithm, using image construct a complete panoramic image mosaicing system, this is the typical representative of image mosaicing technique.

Difference from traditional video surveillance system，panoramic video surveillance system can provide viewers with a complete 360o view． MA Li，ZHANG Mao-jun，XU Wei, etal.[8]designed KD-PVS ，one embedded high resolution panoramic video surveillance system ． They introduces the multiple-camera configuration and video mosaicing algorithm to stitch the video data from multiple camera sources into the panoramic video．KD-PVS system is very convenient for various situations， warehouses， prisons， mobile monitoring ， etc ， especially useful for indoor monitoring． ZHAO Hui and others [9]presented an improved fully-automatic image mosaic algorithm is．By sorting the unordered image sequence and roughly computing the translation offset between adjacent images to speeds up corner match procedure and improves matching stability ,and then using RANSAC algorithm to eliminate outliers to ensure effectiveness of the matched corner pairs， finally using a multi-band blending technique to generate the final panorama．It has less blur or ghost effect after blending，especially when there are noise， moving objects， repeated texture and small overlaps presented in the images．

YIN Run-min，LI Bo-hu, etal. [10] constructed a cylindrical panoramas view of a scene from an image sequence by using multiple image matching algorithms according to self-relation variant after precise focal lens searching adaptively to redraw the accurate overlapping image horizontally and vertically． In order to improve the visual field of photos，FENG Yu-ping，DAI Ming，SUN Li-yue [11] presented an optimized algorithm of automatic image mosaic based on frequency and time domains．By adopting the frequency phase correlation to sort the unordered image sequence and to estimate the overlapping area firstly, and then using the bidirectional greatest correlative coefficient to obtain the initial feature point pairs，and combining the image mean with the linear weight function to implement image mosaic finally, it can efficiently solve the difficulty in confirming corresponding points and achieves the desired visual effect in mosaic images with notable illumination difference

Multiple static cameras or PTZ(Pan-Tilt — Zoom)cameras are often used to monitor activities over a wide area in video surveillance system Miao Li-gang[12] studies the video compositing algorithm of multiple static cameras and video mosaicing algorithm to proposes a key frame based video mosaicing approach for PTZ cameras．these frames are selected based on the amount of overlap and abundance of the texture information．All frames are matched to their latest neighbor key frames，and then back-ground model parameters are updated for the overlapping regions．Key frames have very high alignment accuracy，so it can create accurate background image of the scenes that the moving object has traversed．

From these priors' researches mentioned above, we propose our self-adaptive optimal mosaic line matching algorithm to adjust the size of mosaicing area adaptively with high resolution for real-time monitoring system.

III. DESCRIPTION OF REAL-TIME PANORAMIC VIDEO MOSAICING

Our real-time panoramic video mosaicing system is comprised of three modules: Video Acquisition Module, Images Process Module, and Real-time Update Module. The framework of our system is shown as Fig.1.

The establishment of our system based on the problems we meet on our process of prior images mosaic, the mainly problems are shown as following:(1) Because of the oversize of our cameras' volume, there are external geometric distortions caused by the lack of coincidence in physics amongst each camera viewpoint; (2) Because of the oversize of our cameras' angle and the short-focus, there are internal geometric distortions;(3) Because of the lack of same projection plane amongst the images taken by multiple camera arrays, mosaicing them directly will destroy the visual consistency of actual scenery;(4) Because of the existence of long shot and close shot, there will be ghosts generated for the reason of depth of a field in the real –time process of mosaicing the overlap areas between the adjacent images; (5) Because of the automatic adjust brightness function of camera when meeting inconsistent brightness scenes; there are external optical output image distortions of output image after mosaicing for the different brightness of images taken by the different cameras.

Video Acquisition Module is comprised of 8 camera arrays to get the 360 degree panorama, This camera array put scheme as shown in Fig 1, the height of it is 11cm, by configuring it as octagon with its sides is 10cm, and the angle between each camera is 45°. The focal length of each camera's lens is 3mm, and the each visible angle of these cameras is 90 °. In order to reduce the influence of the size of the camera, we choose common pinhole camera and make our best effort to keep the light heart consistent (as close as possible) and all the cameras in the same horizontal as far as possible.

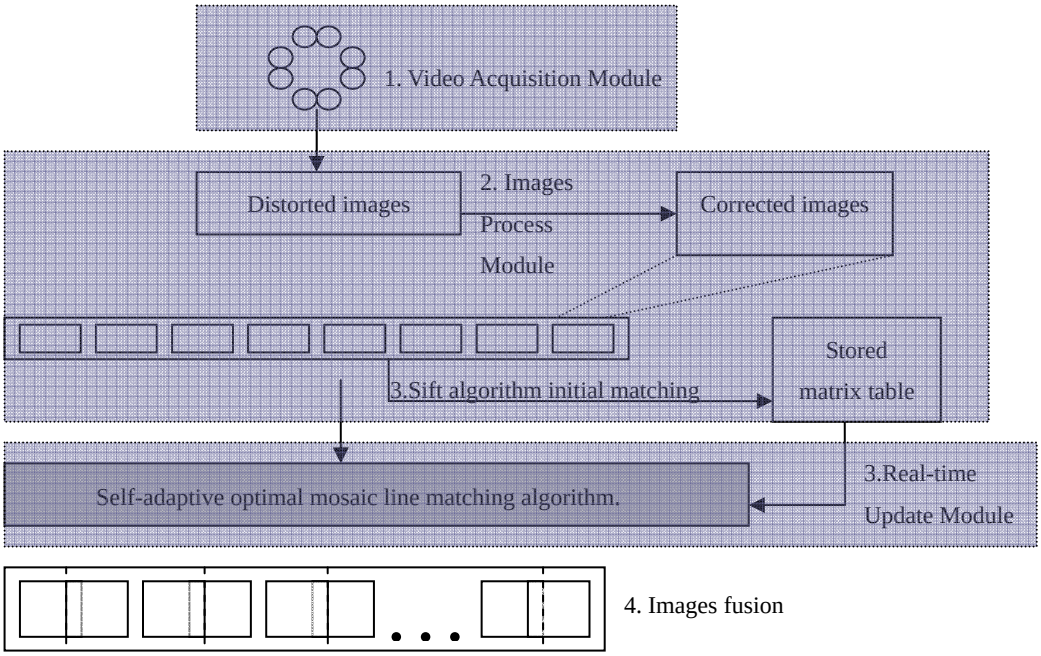


Figure 1. The Framework of Real-time panoramic video mosaicing system

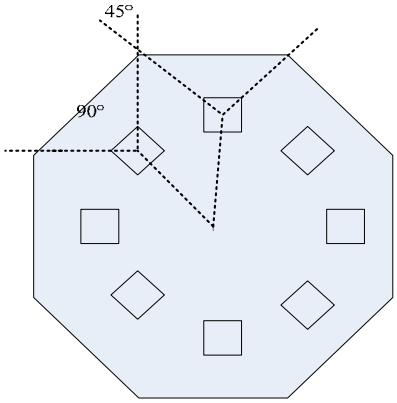


Figure 2. camera arrays

Images Correction Module is used to cylindrical transform the images captured by acquisition module; there are mainly two tasks (or two sub-modules) in this module. The first one is to correct the inconsistent amongst these images caused by some of the cameras' parameters, such as the internal image distortions caused by the oversize view angle of our pinhole cameras.the unprocessed and corrected images are shown as Fig.3 and Fig.4 separately.

This problem must be solved before mosaicing process. Our system adopted the camera calibration deformation correction method mentioned in [13] by Liansheng Sui, Haitao Wang and Jiulong Zhang,etal. The second one is to achieve image pre-matching by SIFT (Scale Invariant Feature Transform) algorithm [14]. In our system, this task is used as initialization stage to process our images. For SIFT algorithm is a mature method, the details of it do not have to be further mentioned in our paper.

The last module is Real-time update module, it is used as real-time image mosaicing stage compared with the SIFT process sub-module. It is designed to improve the defects of unchanged mosaicing line, by the instruction of real-time updating of the mosaicing line, it can resolve the mosaic ghost generated by the inconsistent viewpoint of camera arrays totally. As it is the most important module in our system, we will have a detailed discussion about this module in the next section.

Finally, to make the mosaic area smooth and to enhance image quality, we used a linear weighted fusing method to avoid the intensity disparity and color disparity on the mosaicing line. By adding the image pixel values of the areas' points with a certain weight values, it generates a new image on the overlapping area.

Figure 3. uncorrected image

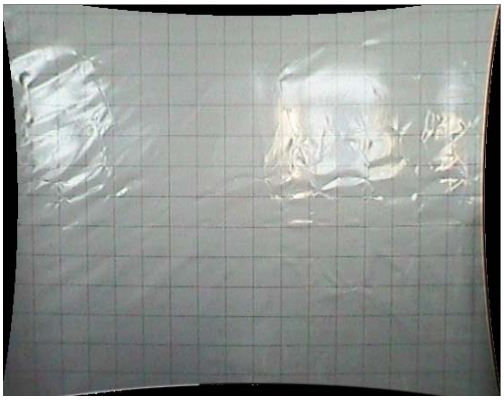


Figure 4. corrected image

IV. SELF-ADAPTIVE OPTIMAL MOSAICING ALGORITHM

A. Significance and Procedures

Theoretically, if all cameras of a video collection system can overlap in a single point of view, the panoramic images mosaicing is just needed in the initial process, i.e., using video collection system to collect the first frame image. Once the geometry parameters amongst the images collected by each camera is computed, and if it is in conformity with the parameters computed in the initial process, the subsequent part can take some mosaics transformation directly.

But there are lots of problems in application, take FlyCam system [4] for an example, for the parameters of the image mosaicing is computed only one time, it can realize real-time images processing speed. But at the same, it will lead to very serious ghost [4] phenomenon. Because of the volume of the camera itself, and for the keep changing distance among cameras and the scenes or objects it takes, it will make the images' geometry parameters constantly change finally. With this in mind, it is necessary to adopt a self-adaptive real-time adjusting algorithm in the process of subsequent video frame mosaicing, which has to consider the mosaic precision and real-time capability at the same time.

After considering various image matching algorithm, we advances a new type of self-adaptive optimal mosaicing algorithm. As mentioned in the section 3, there are two phases on image matching process, the first one is initial matching phase using SIFT algorithm, the second one is real-time image mosaicing stage using self-adaptive optimal mosaicing algorithm, the latter algorithm belongs to one of template matching algorithms .so we can give consideration to both the precision of SIFT algorithm and the simplicity of template matching algorithm.

There are mainly four steps in the initialization process of our system. Firstly, we cylindrical transform the images collected by video acquisition array module to solve the inconsistent visual in the process of image mosaicing. Secondly, we extract the images' SIFT feature points and match them by RANSAC algorithm. Then, we mathematical operate the matched feature points by singular value decomposition to product preliminary affine matrix (short for H matrix) of adjacent images. Finally, we refine H matrix by ML algorithm to calculate the image mosaicing line at this time. So far, the initialization process is finished.

Now the initial matched results amongst adjacent images already are for sure. For image acquisition system with fixed camera, through optical imaging principle and large number of experiments, it is shown that, there is no change on vertical offset of cylindrical image overlapping areas, which means they can be determined by the initial matched results; on the other hand, the horizontal offsets are keeping changing. Therefore, our system uses template matching algorithm, for it has the small amount of calculation, to real-time match and adjust the horizontal offset of image overlapping areas.

B. Algorithm designing

Theoretically, if the camera viewpoint is consistent, the overlap area between adjacent cameras is shown as Fig.5. In this circumstance, long shot or close shot, the overlapping areas have the same proportion of the two adjacent images. It only takes one time to pre-match the two images to make sure the overlapping area between them. But, for there are multiple cameras in our Video Acquisition Module , the problem of inconsistent viewpoint is unavoidable ,shown as Fig.6.Right now, the overlap range is decided the shot are long or close shot, close shot with a narrower overlapping area, long shot with a wider overlapping area, vice versa. There is blind spots existing if the camera is little too close. In the real-time process, for the kept changing overlapping areas, it is necessary to use the self-adaptive mosaicing method to get the new mosaicing area. In this section, we will discuss the designing of our algorithm to real-time self-adaptive optimal mosaicing video stream.

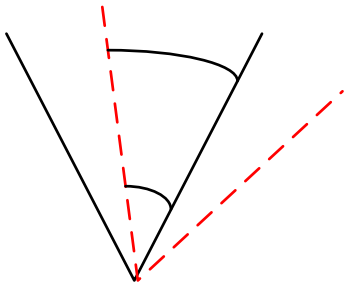


Figure 5. overlap area of consistent viewpoint

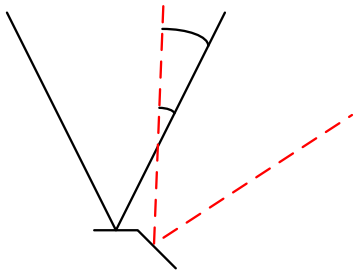


Figure 6. overlap area of inconsistent viewpoint

Algorithm: Self-adaptive optimal mosaicing Algorithm

Input: video streams
Output: the optimal panoramic mosaicing video
1: Correcting images collected by camera arrays
2: Initial image matching stage

2.1 feature values of the images=SIFT(images)

2.2 H matrix=RANSAC(feature values of the images)

$$H=\begin{Bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{Bmatrix}$$

2.3 IML (Initial Mosaicing Line)= Perspective Transformation (H)

$$\begin{cases} x'=\dfrac{m_0x+m_1y+m_2}{m_6x+m_7y+m_8} \\ y'=\dfrac{m_3x+m_4y+m_5}{m_6x+m_7y+m_8} \end{cases} \quad \begin{vmatrix} x,y: \textit{pixel values of } Point_{(x,y)} \textit{on the first images} \\ x',y': \textit{pixel values of } Point_{(x,y)} \textit{ mapped on the second images} \end{vmatrix}$$

2.4 Serializing the table of IML: <L_1,…L_7> to the disk

3: Inputting two images

4:Self-adaptive image matching stage

4.1 taking one line of IML L_1 as the template baseline (L_1 in the left images of the two adjacent ones)

4.2 setting the (IMT)Initial Matching Template based on L_1

4.3 taking the line of IML L_2 as the matching area baseline (L_2 in the right images nearest to L_1')

4.4 setting the TMA (template matching area)based on L_2
TMA =(L_2-△x, L_2+△x);

4.5 for(int i=0;i++;i<10)

4.6 Similarity (x_i)=Similarity(IMT, TMA) ;

$$similarity(IMT,TMA)=\frac{\sum_x\sum_y[f(x,y)-\overline{f}][w(x,y)-\overline{w}]}{\{\sum_x\sum_y[f(x,y)-\overline{f}]^2\sum_x\sum_y[w(x,y)-\overline{w}]^2\}^{1/2}}$$

$$f(x,y):\ pixel\ values\ of\ point_{(x,y)}\ on\ IMT\ \ \overline{f}=Avg(\sum_{(xi,yi)\in IMT} f(x_i,y_i))$$

$$w(x,y):\ pixel\ values\ of\ point_{(x,y)}\ on\ TMA\ \ \overline{w}=Avg(\sum_{(xi,yi)\in TMA} w(x_i,y_i))$$

4.7 argMax (△x_i)=Max(Similarity),

4.8 Return △x_i

4.9 L_2= L_2+△x_i

4.10Return L2

5: Return to the step 3 to take the next two images.

6: Getting the optimal mosaicing line of the adjacent images as IML<L1',…L7' >

7: Setting the OMA (optimal matching area) based on the optimal IML

8: Fusing Image:

$$\begin{bmatrix} r_{(x'',y'')} \\ g_{(x'',y'')} \\ b_{(x'',y'')} \end{bmatrix}=d\begin{bmatrix} r_{(x',y')} \\ g_{(x',y')} \\ b_{(x',y')} \end{bmatrix}+(1-d)\begin{bmatrix} r_{(x,y)} \\ g_{(x,y)} \\ b_{(x,y)} \end{bmatrix} \ where\ d\in(0,1)\ and \begin{vmatrix} (r_{(x,y)},g_{(x,y)},b_{(x,y)}): RGB\ of\ Point_{(x,y)} on\ IMT, \\ (r_{(x',y')},g_{(x',y')},b_{(x',y')}): RGB\ of\ Point_{(x',y')}\ on TMA, \\ (r_{(x'',y'')},g_{(x'',y'')},b_{(x'',y'')}): RGB\ of\ Point_{(x'',y'')}\ on OMA, \end{vmatrix}$$

9: Return optimal mosaicing images

10: Taking the next frame of the input video streams

11: Iterating the step3 to step 9 to the last frame of the video streams

There is another problem about our proposed algorithm. At line 4.5 of the algorithm description, we actually search along the horizontal direction(that is x-direction). However, the most generic case is to search along the line L2, which is defined by perspective transformation (its formula is at line 2.3). In other words, in most general case, the line L2 is not necessary horizontal, so only searching along horizontal direction will sometimes fail to find correspondence. In a sum, my algorithm is only suitable for the specific case that only translation exists between adjacent images and rotation between them is not allowed.

However, there is no scaling and rotation relationship amongst output images of our cameras cluster in our system, it is suitable for the plan of engineering design, although not comprehensive enough in terms of theory

V. RESULTS

Experiments are conducted on a computer with Pentium 4, 2.4GHz CPU, 1GB RAM based on VC 6.0 program, and eight pinhole cameras cluster. By taking eight primary images with the resolution of 480*260, we can get the panoramic mosaicing image with the resolution of 3580*480 at the mosaicing speed of 25fps.

By the SIFT algorithm, we can get the geometric relationship between the adjacent cameras, which is shown as Tab.1. <m0,…, m8> are the parameters of the H matrix. The content under 'The i-th camera' means the values of number i camera in our panoramic camera cluster, the range of i is from 1 to 7, for the values of 8-th camera can be determined by the former seven cameras. The x-offset is horizontal offset between the adjacent images. The Y-offset is vertical offset between the adjacent images.Fig.7 is the picture of our panoramic camera cluster. The panoramic mosaicing images are shown as Fig.8, Fig.8 and Fig.10; they are the 20-th, 150-th and 320-th frame of the panoramic mosaicing video streams separately.

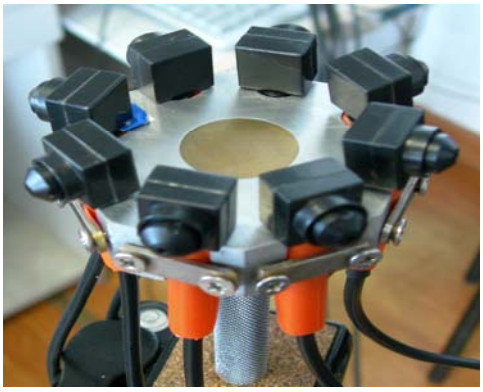


Figure 7. Panoramic camera cluster

Table.I Geometric relationship between the adjacent cameras

H	The i-th camera						
matrix	1	2	3	4	5	6	7
m0	0.988	0.978	0.953	0.8960	0.931	0.971	0.8356
m1	-0.026	0.071	-0.031	-0.018	-0.021	-0.018	-0.0328
m2	-244.033	-318.886	-336.569	-308.124	-342。178	-331.481	-271.874
m3	-0.023	-0.004	-0.059	-0，031	-0.045	-0.061	-0.046
m4	0，987	0.796	0.930	0.951	0.871	0.894	0.9713
m5	-10.749	1.966	-25.312	-23.418	-21.596	1.851	-15.6478
m6	0	-0.000	0	-0.0001	0	-0.0001	0
m7	0	0	0	0	0	0	0
m8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
X-offset	239	150	127	122	104	134	129
Y-offset	1	1	-2	0	-1	0	1

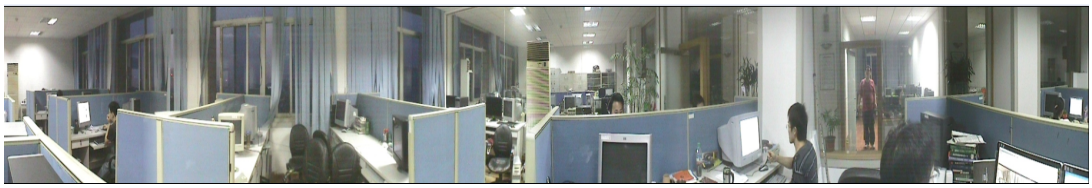


Figure 8. The 20-th frame of the panoramic mosaicing video streams

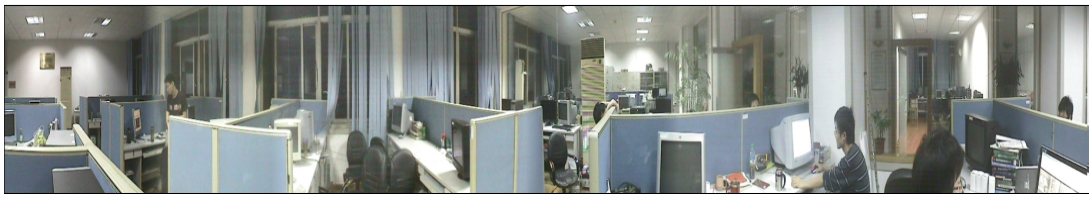


Figure 9. The 150-th frame of the panoramic mosaicing video streams

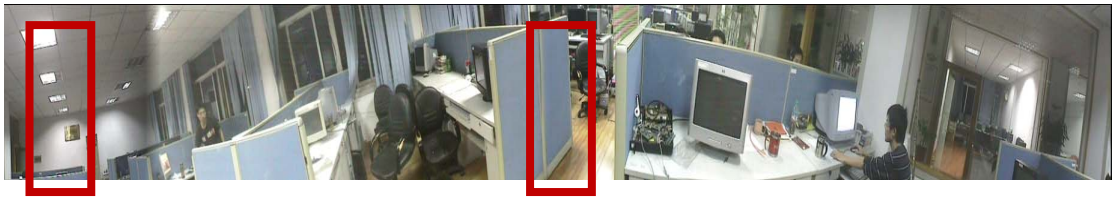


Figure 10. The 320-th frame of the panoramic mosaicing video streams

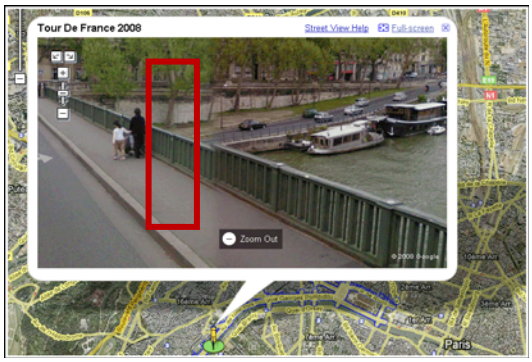


Figure 11. The Point Grey's LadyBug2

The Fig.11 is an image outputted form designed method by Point Grey's LadyBug2 Cop., the technology of which is used by GoogleMap panoramic image mosacing. By the comparison of blurring upon zooming into the red boxes in Fig.11 and Fig.10, ghost in the former one and nonexistent in the other one, we can draw a conclusion that our method has better performance in panoramic image mosacing.

VI. CONCLUSION

To solve the problems in the real-time monitoring systems, such as the field vision is too small to capture, etal. We designed this real-time panoramic monitoring system, which can output the panoramic video signals in a range of 360 degree. Firstly, we use the improved SIFT algorithm to compute the geometric relationship between the adjacent cameras for pre-matching. Facing the problem of inconsistent viewpoint, we take the self-adaptive template matching method to adjust the images matching at every stage or moment, which make our system very suitable for the application which requests high real-time performance.

Our algorithm can settle the ghost phenomenon caused by the constantly changing depth of a field, and with the help of linear weighted fusing method, we can eliminate the intensity disparity resulted from the different cameras. The entire above-mentioned designing make our system has the characteristics of good real-time, high resolution and ideal effect of video mosaicing.

REFERENCE

[1] Vic Nalwa, PhD. Outwardly Pointing Cameras. Fellow of the IEEE President, FullView, Inc.

[2] Vishvjit S. Nalwa. A True Omni-Directional Viewer [J].Vishvjit S. NalwaT&T Bell AT&T Bell Laboratories. Holmdel, NJ 07733, U.S.A.

[3] Jonathan Foote , Don Kimber. FlyCam: Practical Panoramic Video and Automatic Camera Control[J].2000.ICME 2000.2000 IEEE International Conference on Multimedia and Expo, Vol. 3,30July pp:1419-1422

[4] Heung-Yeung Shum, Richard Szeliski·Construction of panoramic image mosaics with global and local alignment [J]· International Journal of Computer Vision, 2000, 36(2): 101~130

[5] Fang xian-yong,Pan Zhi-geng,Xu Dan,An improved algorithem for image mosaic ,JOURNAL OF COMPUTER-AIDED DESIGN & COMPUTER GRAPHICS， Nov.,2003 Vol.l5，No.11, 2003.11

[6] CAI Li-huan, LIAO Ying-hao, GUO Dong-hui. Study on Image Stitching Methods and Its Key Technologies [J]. Computer Technology and Development, 2008, 18(3):1-4.

[7] Richard Szeliski. Video Mosaics for Virtual Environments [J].IEEE COMPUTER GRAPHICS AND APPLICATIONS. 1996, 16(2):pp. 22-30

[8] MA Li，ZHANG Mao-jun，XU Wei， A High Resolution Panoramic Video Monitoring System Based on Video Mosaicin，Journal of Image and Graphics，V01．13，No．12，Dec．，2008

[9] ZHAO Hui，CHEN Hui，YU Hong，An Improved Fully-automatic Image Mosaic Algorithm， Journal of Image and Graphics，V01.12．No.2,Feb,2007

[10] YIN Run-min，LI Bo-hu，CHAI Xu-dong, Adaptive Cylindrical Panoramas Mosaic, Journal of Image and Graphics，V01.13，No.6,June，2008

[11] FENG Yu-ping，DAI Ming，SUN Li-yue，ZHANG Wei,Optimized design of automatic image mosaic,Optics and Precision Engineering, V01．18 No.2,Feb．20lO

[12] Miao Li-gang .Image mosaicing and compositing algorithm for video surveillance ,Chinese Journal of Scientific Instrument V01．30 No.4,Apr．2009

[13] Liansheng Sui, Haitao Wang and Jiulong Zhang A New Method of Image Rectification Using Jacobian Determinant, Proceedings of the 2008 IEEE International Conference on Information and Automation June 20 -23, 2008, Zhangjiajie, China

[14] David G.Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, Vol.60, Page90-110, Nov 2004.

Lin Zeng, JiangXi, China, 1983,
Bachelor: Electronic Information School, Wuhan University, 2006
Second-year PhD student: Image Process, Electronic Information School, Wuhan University, 2009

Xi Chen, Hubei**,** China, 1980, Associate Professor, Institute of Microelectronics and Information Technology Wuhan University.
Interest research: Signal Process, Artificial Intelligence

Dexiang Deng, Hubei, China, 1961, Professor, Electronic Information School, Wuhan University,
Research interest: Spatial Image Process

A Meta-learning-based Approach for Detecting Profile Injection Attacks in Collaborative Recommender Systems

Fuzhi Zhang
School of Information Science and Engineering, Yanshan University, Qinhuangdao, China
Email: xjzfz@ysu.edu.cn

Quanqiang Zhou
School of Information Science and Engineering, Yanshan University, Qinhuangdao, China
Email: zhouqiang128@126.com

***Abstract*—Recent research has shown the significant vulnerabilities of collaborative recommender systems in the face of profile injection attacks, in which malicious users insert fake profiles into the rating database in order to bias the system's output. A single Support Vector Machine (SVM) approach for the detection of profile injection attacks, however, suffers from low precision. With this problem in mind, in this paper we propose a meta-learning-based approach to detect such attacks. In particular, we propose an algorithm to create the diverse base-level training sets through flexible combination of various attack types. Combining the created training sets with SVM, we construct the base-level and meta-level classifiers. Based on these classifiers, we present a meta-learning-based detection algorithm which uses the meta-classifier to integrate the outputs of the base-classifiers and generates the final results of detection. The diversities among the base-classifiers effectively reduce the correlation of the misclassifications and improve the predictive capability of the meta-level. We conduct comparative experiments with a single SVM and the voting-based ensemble method on different-scale MovieLens datasets. The experimental results show that the proposed approach can effectively improve the precision under the condition of holding a high recall.**

***Index Terms*—meta-learning, support vector machine, profile injection attacks, attack detection, collaborative filtering recommendation**

I. INTRODUCTION

Collaborative filtering recommender systems [1] can filter out the information to satisfy the users' interest according to the established user profiles and actively recommend the information to users. It provides an effective way to solve the information overload problem on the Internet and it has become an important part of many e-commerce sites. Due to its natural openness, however, some malicious users artificially inject a large number of fake profiles into the system in order to bias the recommendation results to their own advantage. These attacks, where a malicious user inserts fake profiles into the rating database to influence the system's recommendation behavior, have been termed "shilling" attacks or "profile injection attacks" [2]. To distinguish the genuine profiles, we usually call the fake profiles as attack profiles. Attack model is an approach for attackers to construct the attack profiles according to the knowledge about the recommender system's rating database, products, users, etc. According to its construction strategy, the attack model can be divided into the random attack, bandwagon attack, average attack, segment attack, etc [3-5]. For the different purposes of attacks, profile injection attacks can be divided into push attack and nuke attack [6], which increase and decrease the recommendation frequency of the target item respectively. The strength of profile injection attacks is measured by attack size and filler size [7]. Attack size is the ratio between the number of attack profiles and the number of genuine profiles in a recommender system. Filler size is the ratio between the number of ratings in an attack profile and the number of all items in the recommender system. In the face of the various attack types, therefore, how to effectively identify and resist profile injection attacks has become an urgent problem to be solved for the well development and extensive application of collaborative filtering recommender systems.

Recently, the detection of profile injection attacks has become a hot research area in collaborative filtering recommender systems. Chirita et al. [8] proposed several statistical features to describe attack profiles for the detection of high-density attack profiles. Su et al. [9] developed a similarity spreading algorithm to detect simple groups of attack profiles. Mehta et al. [10-12] presented a PCA-based detection algorithm by using Principal Component Analysis (PCA) technique to filter out the attack profiles. PCA-based algorithm is effective

Manuscript received May 2, 2011; revised August 2, 2011; accepted August 9, 2011.
Supported by the Natural Science Foundation of Hebei Province, China (F2011203219) and the Special Fund for Fast Sharing of Science Paper in Net Era by CSTD of Ministry of Education, China (20101333110013).

doi:10.4304/jcp.7.1.226-234

for detecting various attack types. But it needs to know the total number of attack profiles injected into the rating database of the recommender system before detecting, which is difficult to estimate in practice. Bryan et al. [13] proposed a detection algorithm, which is called UnRAP, to identify the attack profiles by introducing the variance adjusted mean square residue. This algorithm can detect random attack and average attack successfully, but it can not detect bandwagon attack. Hurley et al. [14] designed supervised and unsupervised Neyman-Pearson detectors based on statistical detection theory. These detectors can only detect the attack profiles which have the specific distributions. He et al. [15] used rough set theory to detect attack profiles. They simply take the feature values of user profiles as the condition attributes of the decision table to perform the operations of reducing data and generating rules. This method can detect most of the attack profiles, but the precision is low. Burke et al. [16-18] trained three supervised classifiers (i.e. KNN, C4.5 and SVM) to detect the attacks by extracting the features of user profiles. Of the three classifiers, SVM has the best detection performance. To effectively identify the various attack profiles in practice, which are constructed by various attack models at various attack sizes and filler sizes, the training set of SVM needs to contain enough attack samples. Although this operation can ensure that the classifier has the capability of detecting most of the attack profiles in the recommender system, the classifier's precision is poor due to the fact that too many genuine profiles are misclassified as attack profiles. In our previous work, we presented an improved PCA-based detection algorithm by introducing the normal cloud model theory and explored its effectiveness when detecting profile injection attacks with small attack sizes [19][20]. In [21], we proposed an anomaly detection algorithm based on analyzing rating distribution characteristics of item over rating time series and we showed our algorithm to be more effective at detecting target items than the method described in [22].

In this paper, we propose a meta-learning-based approach to improve the precision of detecting profile injection attacks. Our contributions mainly include:

(1) We propose an algorithm to create diverse base-level training sets;
(2) We present a meta-learning-based detection algorithm to improve the overall performance of attack detection;
(3) We conduct simulation experiments on different-scale MovieLens datasets. The experimental results show that the proposed approach can not only hold a high recall, but also effectively improve the precision of the detection.

II. THEORETICAL FOUNDATION

Meta-learning is an ensemble learning approach. The underlying idea of this approach is based on relearning the existed knowledge to boost overall predictive effectiveness. The meta-classifier (or level-1 model) tries to acquire how the outputs of the base-classifiers (or level-0 models) should be combined to obtain the final classification [23].

Stacked Generalization [24] is one of the most commonly used meta-learning methods. The details of this method are given as follows.

Given a dataset $\mathcal{S} = \{(y_n, \boldsymbol{x}_n) \mid n = 1, 2, ..., N\}$, where y_n is the class value and $\boldsymbol{x}_n$ is a vector representing the attribute values of the nth instance, N is the number of instances in $\mathcal{S}$, this method randomly splits the dataset into J almost equal parts $\{\mathcal{S}_1, ..., \mathcal{S}_J\}$. $\mathcal{S}_j$ and $\mathcal{S}^{(-j)} = \mathcal{S} - \mathcal{S}_j$ are defined to be the test and training set for the jth fold of a J-fold cross-validation. $\mathcal{S}^{(-j)}$ is used to train K classification algorithms $\{\mathcal{A}_1, ..., \mathcal{A}_K\}$ to generate K classification models $\{\mathcal{M}_1, ..., \mathcal{M}_K\}$ respectively, where $\{\mathcal{A}_1, ..., \mathcal{A}_K\}$ are called level-0 algorithms, $\{\mathcal{M}_1, ..., \mathcal{M}_K\}$ are called level-0 models or base-classifiers.

For each instance $\boldsymbol{x}_j$ in $\mathcal{S}_j$, let $\{\mathcal{M}_1(\boldsymbol{x}_j), ..., \mathcal{M}_K(\boldsymbol{x}_j)\}$ denote the predictions of the models $\{\mathcal{M}_1, ..., \mathcal{M}_K\}$ on $\boldsymbol{x}_j$, then, all the predictive results create a new dataset:

$$\mathcal{S}_{meta} = \{(y_j, \mathcal{M}_1(\boldsymbol{x}_j), \ldots, \mathcal{M}_K(\boldsymbol{x}_j)) \mid j = 1, \ldots, L\} \quad (1)$$

where L is the number of instances in $\mathcal{S}_{meta}$. $\mathcal{S}_{meta}$ is called level-1 training set or meta-level training set. $\mathcal{S}_{meta}$ is used to train a learning algorithm to generate a classification model $\mathcal{M}_{meta}$, where $\mathcal{M}_{meta}$ is called level-1 model or meta-classifier, the learning algorithm is called level-1 generalization algorithm.

Given a new instance q, the level-0 models produce a vector $(\mathcal{M}_1(q), ..., \mathcal{M}_K(q))$. This vector is input to the level-1 model $\mathcal{M}_{meta}$, whose output is the final predictive result for q.

III. META-LEARNING-BASED APPROACH FOR ATTACK DETECTION

To facilitate the discussions in this section, we introduce some symbols: $\mathbb{I}$ is the set of items in the recommender system, $\boldsymbol{x}$ is a rating vector in a user profile and $\boldsymbol{x} = (rating_1, ..., rating_i, ..., rating_{\mathbb{I}})$, where $rating_i \in \varnothing \bigcup \{rating_{min}, ..., rating_{max}\}$, $y_n \in \{-1, 1\}$ is the class value of the nth instance, where -1 represents a genuine profile and 1 represents an attack profile. $f_k(\boldsymbol{x}) = \mathrm{SVM}_k(\boldsymbol{x})$ denotes the prediction of the kth classifier SVM_k on $\boldsymbol{x}$.

The framework of the meta-learning-based approach for detecting profile injection attacks is depicted in Figure 1. The detection model of meta-learning is in the dashed box. This model contains two training processes which are base-level training and meta-level training. Both levels use SVM as their learning algorithms. The rating database and the attack profiles are input to the proposed algorithm for creating the base-level training sets.

$\{f_1(\boldsymbol{x}),...,f_K(\boldsymbol{x})\}$, which are the predictions of the K base-classifiers $\{\text{SVM}_1,...,\text{SVM}_K\}$ on $\boldsymbol{x}$, are input to SVM_{meta}. The final predictive results of the test set are generated by SVM_{meta}.

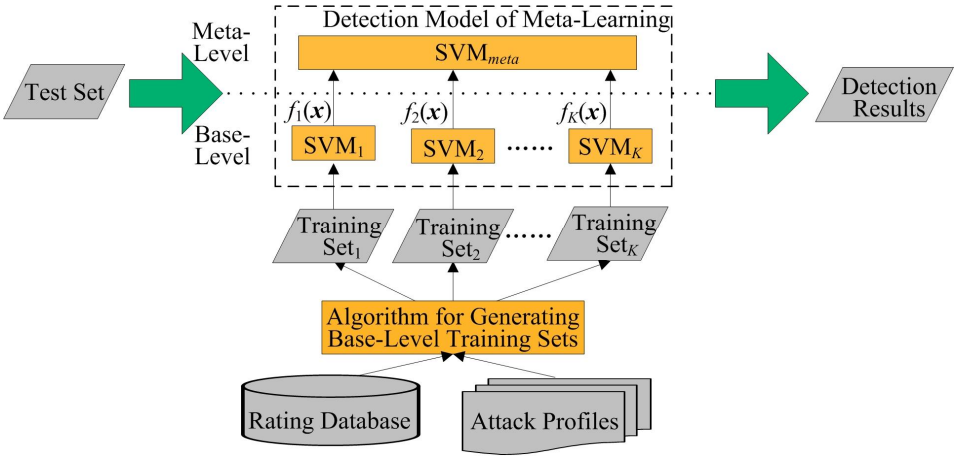


Figure 1. Framework of the meta-learning-based approach for detecting profile injection attacks

A. *Construction of Base-level Classifiers*

To improve the predictive quality of the meta-level model, the base-classifiers have to be diverse [25]. Stacked Generalization uses the strategy of cross-validation to create the base-training sets. Although this strategy can create different base-training sets, serious skew may be introduced to these sets between the attack profiles and the genuine profiles. In the skewed training sets, the training samples can not accurately reflect the data distributions of the entire space and the classifiers trained from these sets would be weak [26]. To balance the proportion of the training samples between attack profiles and genuine profiles, we propose an algorithm to create the diverse base-level training sets through flexible combinations of various attack models at various attack sizes and filler sizes.

Let $\text{A}=\{a_1\%,...,a_P\%\}$ denote P different attack sizes, $\text{B}=\{b_1\%,...,b_R\%\}$ denote R different filler sizes, $\text{C}=\{c_1,...,c_W\}$ denote W different attack models and i_{target} denote the target item. Let quad $(a\%,b\%,c,i_{target})$ denote the set of attack profiles which are constructed by attack model c at attack size of $a\%$ across filler size of $b\%$ and their target item is i_{target}. Let getTargetItem(Ⅰ) be a function to randomly select an item from Ⅰ. Let $\text{T}=\{(y_n,\boldsymbol{x}_n)\,|\,n=1,2,\ldots,N\}$ denote the rating database of the recommender system, where N is the number of user profiles in T which dose not contain attack profiles. Let $K=P\times R$ denote the number of base-training sets and Z denote the set of base-training sets. The algorithm for creating the base-training sets is described as follows:

Algorithm 1 Flexible method for creating the base-training sets
Input: A, B, C, P ,R ,W, T, Ⅰ, i_{target}, K
Output: Z
1:Z← ∅
2:Mark← ∅
3:**For** k = 1 **to** K **Do**
4: $\text{T}_{Attack\ Profile}$← ∅
5: $\text{T}_{Genuine\ Profile}$← ∅
6: flag←0
7: **For** p = 1 **to** P **Do**
8: **For** r = 1 **to** R **Do**
9: **For** w = 1 **to** W **Do**
10: i_{target}←getTargetItem(Ⅰ)
11: T←T ∪ $\{(y_e, \boldsymbol{x}_e)|\ \boldsymbol{x}_e \in$ (a%, b%, c, i_{target})$\}$
/* Insert attack profiles into the rating database */
12: $\text{T}_{Attack\ Profile}$←$\text{T}_{Attack\ Profile}$ ∪ $\{(y_e, \boldsymbol{x}_e')|\ \boldsymbol{x}_e'$ is $\boldsymbol{x}_e$ after features extraction of T$\}$
/* Create samples of attack profiles */
13: **If**((p,r,w) ∉ Mark) **and** (flag < W)) **Then**
14: $\text{T}_{Genuine\ Profile}$←$\text{T}_{Genuine\ Profile}$ ∪ $\{(y_n, \boldsymbol{x}_n')|\ \boldsymbol{x}_n'$ is $\boldsymbol{x}_n$ after features extraction of T$\}$
/* Create samples of genuine profiles */
15: Mark←Mark ∪ $\{(p,r,w)\}$
16: flag←flag+1
17: **End If**
18: T←T-$\{(y_e, \boldsymbol{x}_e)|\ \boldsymbol{x}_e \in$ (a%, b%, c, i_{target})$\}$
19: **End For**
20: **End For**
21: **End For**
22: Z←Z ∪ $\{\text{T}_{Attack\ Profile} \cup \text{T}_{Genuine\ Profile}\}$
23:**End For**
24:**Return** Z

In the stage of features extraction for user profiles, we use the detection attributes described in [17] to extract 13 features as follows:

(1) Six generic features are WDMA, RDMA, WDA, Length Variance, DegSim (k=450) and DegSim′ (k=2, d=963) respectively. These features attempt to make an attack profile look different from a genuine profile through capturing some of the characteristics.
(2) Seven attack model features are FMD (random attack, push), FAC (random attack, push), FMD

(bandwagon attack, push), FAC (bandwagon attack, push), FMV (average attack, push), FMD (average attack, push) and PV (average attack, push) respectively. These features are designed to recognize the characteristics of known attack models.

Let $\{V_1(\boldsymbol{x}_h),...,V_{13}(\boldsymbol{x}_h)\}$ denote the feature values of $\boldsymbol{x}_h$ in a base-training set. The kth base-training set is:

$$\mathrm{z}_k = \{(y_h, V_1(\boldsymbol{x}_h),...,V_{13}(\boldsymbol{x}_h)) \mid h = 1,\ldots,H\} \quad (2)$$

where H is the number of samples in z_k.

All the base-training sets are as follows:

$$\mathrm{Z} = \{\mathrm{z}_1,\ldots,\mathrm{z}_K\}. \quad (3)$$

The base-level SVMs learn the base-training sets respectively in Z to generate the base-classifiers $\{\mathrm{SVM}_1,...,\mathrm{SVM}_K\}$.

In Algorithm 1, the base-training sets still contain enough samples of attack profiles in order to effectively identify the various attack profiles. That is to say, we do not attempt to reduce the misclassifications of the base-classifiers, but to create the diverse base-training sets under the condition that the recommender system is attacked by various attack types. These base-training sets can increase the diversities of the base-classifiers, as well as reduce the correlations of the misclassifications. Furthermore, the predictive capability of the meta-level can be improved by using these diverse base-classifiers.

B. Construction of Meta-level Classifier

To create the meta-training set, we set K=1 in Algorithm 1 to create a base-training set:

$$\mathrm{z} = \{(y_l, V_1(\boldsymbol{x}_l),...,V_{13}(\boldsymbol{x}_l)) \mid l = 1,...,L\} \quad (4)$$

where L is the number of samples in z.

The set z is input to the base-classifiers $\{\mathrm{SVM}_1,...,\mathrm{SVM}_K\}$, whose outputs are used as the meta-training set:

$$\mathrm{z}_{meta} = \{(y_l, f_1(\boldsymbol{x}_l),..., f_K(\boldsymbol{x}_l)) \mid l = 1,...,L\}. \quad (5)$$

We still select SVM as the learning algorithm of the meta-level. SVM has a good generalization performance and it can find a decision surface that optimally separates the instances into two classes based on the Structural Risk Minimization Principle. The decision function is as follows [27]:

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{M} y_i a_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b\right) \quad (6)$$

where, M is the number of training samples, $\boldsymbol{x}$ is the vector of a test sample, $\boldsymbol{x}_i$ and y_i denote the vector and class value of the ith training sample, $K(\boldsymbol{x}, \boldsymbol{x}_i)$ is a kernel function, a_i and b are the parameters of the model. a_i can be learned by solving following quadratic programming problem:

$$\max Q(a) = \sum_{i=1}^{M} a_i - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} a_i a_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$s.t.\ \sum_{i=1}^{M} a_i y_i = 0, 0 \le a_i \le C, i = 1,\ldots,M. \quad (7)$$

Since the outputs of the base-classifiers are binary, a lot of input data of the meta-level are reduplicative. The special advantages of SVM are that SVM is only sensitive to the data on the boundary and it only uses the support vectors to predict the class value for a new instance. The existence of the reduplicative data does not affect the classification performance of this algorithm.

In the end, the meta-level SVM learns z_{meta} to generate the meta-classifier SVM_{meta}.

C. Meta-learning-based Algorithm for Attack Detection

Based on the constructed classifiers $\{\mathrm{SVM}_1,...,\mathrm{SVM}_K\}$ and SVM_{meta}, we present an algorithm to detect profile injection attacks.

Let $\mathrm{T}_{test} = \{(y_q, \boldsymbol{x}_q) \mid q = 1,...,Q\}$ denote the test set, where Q is the number of user profiles in T_{test} and y_q is unseen to the classifiers. Let T_{result} denote the set of the final detection results. The algorithm for detecting profile injection attacks is described as follows:

Algorithm 2 The meta-learning-based algorithm for attacks detection

Input: T_{test}

Output: T_{result}

1:$\mathrm{T}_{result} \leftarrow \varnothing$
2:$\mathrm{T}_{test}' \leftarrow \varnothing$
3:$\mathrm{T}_{test}' \leftarrow \{(y_q, \boldsymbol{x}_q') \mid \boldsymbol{x}_q'$ is $\boldsymbol{x}_q$ after features extraction of $\mathrm{T}_{test}\}$
4:**For** q=1 **to** Q **Do**
5: **For** k=1 **to** K **Do**
6: $f_k(\boldsymbol{x}_q) \leftarrow \mathrm{SVM}_k(\boldsymbol{x}_q')$
/*Generate the outputs of the base-classifiers */
7: **End For**
8: $f_{meta}(\boldsymbol{x}_q) \leftarrow \mathrm{SVM}_{meta}(f_1(\boldsymbol{x}_q),\ldots,f_K(\boldsymbol{x}_q))$
/* Generate the output of the meta-classifier */
9: $\mathrm{T}_{result} \leftarrow \mathrm{T}_{result} \cup \{f_{meta}(\boldsymbol{x}_q)\}$
10:**End For**
11:**Return** T_{result}

In Algorithm 2, the base-classifiers firstly generate their classification results. Then, the meta-classifier integrates these results and outputs the final predictions for the test set. Since each base-classifier is obtained using different knowledge, the hypothesis space is explored differently. Thus, the correlations of the misclassifications are effectively reduced. The combination in the meta-level performs better than the base-classifier and it also performs better than the simple voting-based ensemble method. In our experiments below, we show support for this intuition.

IV. EXPERIMENTS AND EVALUATIONS

A. Experimental Data and Settings

We select two different-scale MovieLens[1] datasets as the experimental data in this paper:

(1) MovieLens 100K dataset. This dataset consists of 100,000 ratings on 1,682 movies by 943 users. All ratings are integer values between 1 and 5, where 1 is the lowest (disliked) and 5 is the highest (most liked). Each user in this dataset has rated at least 20 movies. We partition the dataset in half. The first half is used as the parameter T in Algorithm 1 to create base-training sets. To create each test set, the second half is injected with various attack profiles.

(2) MovieLens 1M dataset. This dataset consists of 1,000,209 ratings on 3,900 movies by 6,040 users. All ratings are integer values between 1 and 5, where 1 is the lowest (disliked) and 5 is the highest (most liked). We randomly select 1,000 user profiles from this dataset as the parameter T in Algorithm 1 to create base-training sets. To create each test set, we randomly select 1,000 user profiles to form a new set from the remaining dataset and then inject various attack profiles into this new set.

In the stage of training, we set A={1%, 2%, 5%, 10%}, B={1%, 3%, 5%, 10%, 25%}, C={random attack, bandwagon attack, average attack} in Algorithm 1 to create the base-training sets. Then, we combine these sets with the methods described in section III.*A* and section III.*B* to generate the base-classifiers and the meta-classifier. These classifiers are generated using libsvm[2].

To create the test sets, the attack profiles, which are constructed by the attack models of random, bandwagon and average attack at filler sizes of 3% and 5% across attack sizes of 1%, 2%, 5% and 10%, are individually injected into the second half of MovieLens 100K dataset and the second 1,000 user profiles of MovieLens 1M dataset described above. We randomly select 50 movies as the target items for each test. Each of these movies is attacked individually and the average is reported for all experiments. So, the final metric values in section IV.*C* and section IV.*D* are the average values of these experiments.

We only detect the push attack in this paper. Through changing the ratings of the target item from maximum to minimum, we can use the same method to detect the nuke attack.

B. Evaluation Metrics

To measure the detection performance, we use the standard measurements of recall and precision [18]:

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{8}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

where, TP is the number of attack profiles correctly detected, FP is the number of genuine profiles misclassified as attack profiles, FN is the number of attack profiles misclassified as genuine profiles.

C. Experimental Results and Analysis on MovieLens 100K Test Sets

To verify the effectiveness of the proposed approach, the test sets are detected individually by three detection method which are Algorithm 2 (called Meta SVM), the method of a single SVM (called single SVM) in [17] and the voting method (called Voting SVM) which ensembles the outputs of the base-classifiers through majority voting method [28].

The detection results of these three methods on the MovieLens 100K test sets are shown in Table Ⅰ, Table Ⅱ, Figure 2 and Figure 3.

TABLE Ⅰ.

RECALL ON MOVIELENS 100K TEST SETS FOR VARIOUS ATTACK MODELS AT FILLER SIZE=3% ACROSS VARIOUS ATTACK SIZES

Attack Model	Random Attack				Bandwagon Attack				Average Attack			
Attack Size	1%	2%	5%	10%	1%	2%	5%	10%	1%	2%	5%	10%
Single SVM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Voting SVM	0.98	0.98	0.99	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00
Meta SVM	0.99	0.99	1.00	1.00	1.00	0.99	0.99	0.99	0.98	0.99	0.99	0.99

TABLE Ⅱ.

RECALL ON MOVIELENS 100K TEST SETS FOR VARIOUS ATTACK MODELS AT FILLER SIZE=5% ACROSS VARIOUS ATTACK SIZES

Attack Model	Random Attack				Bandwagon Attack				Average Attack			
Attack Size	1%	2%	5%	10%	1%	2%	5%	10%	1%	2%	5%	10%
Single SVM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Voting SVM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Meta SVM	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99

[1]http://www.grouplens.org/node/73
[2]http://www.csie.ntu.edu.tw/~cjlin/libsvm

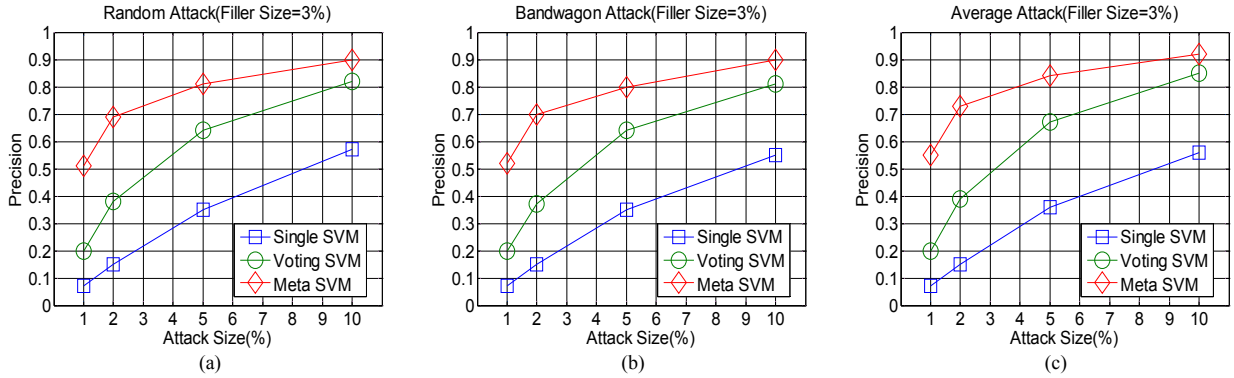


Figure 2. Precision on MovieLens 100K test sets for various attack models at filler size=3% across various attack sizes

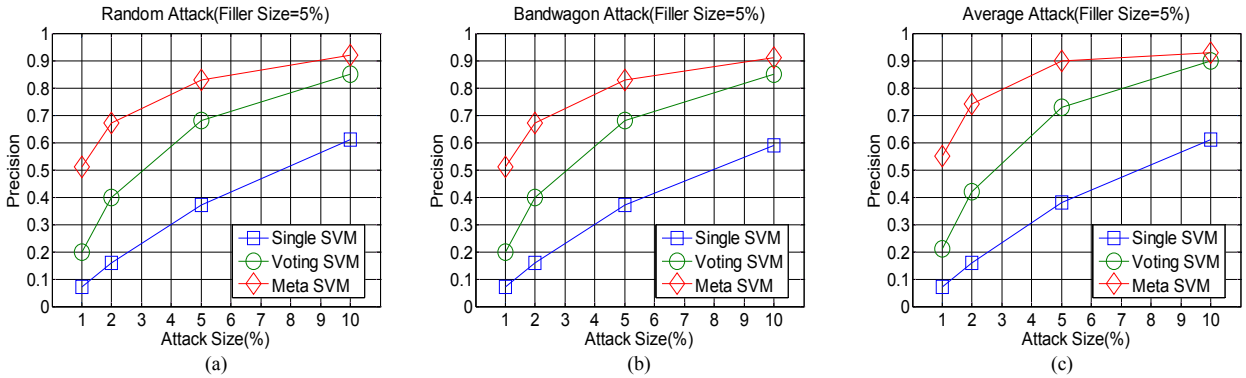


Figure 3. Precision on MovieLens 100K test sets for various attack models at filler size=5% across various attack sizes

As shown in Table Ⅰ and Table Ⅱ, most recalls of Meta SVM are between 0.99 and 1. These results, which are similar to the results of Single SVM and Voting SVM, keep at a high level. This is easy to explain, since each base-classifier itself can detect most of the attack profiles, the integration of these classifiers can still own this capability.

In terms of the attack effects on recommendations, it is discovered that the average attack is more powerful than others [3][4]. However, as shown in Table Ⅰ and Table Ⅱ, Meta SVM is able to effectively identify average attacks at various filler sizes across attack sizes and the recall can reach 0.99. The reason for these high recalls is that the useful classification information of the supervised classifier focuses on the training set where we had injected enough attack samples.

Figure 2 and Figure 3 show that the precision of Meta SVM has been significantly improved compared to the precision of Single SVM and Voting SVM. At 1% attack size, the average precision of Meta SVM increases by 6.6 percentage points and 1.7 percentage points compared to the average precision of Single SVM and Voting SVM respectively. At 10% attack size, the average precision of Meta SVM reaches 0.91 while the average precisions of Single SVM and Voting SVM are 0.58 and 0.85 respectively. These results illustrate the success of Meta SVM in reducing the misclassifications.

D. Experimental Results and Analysis on MovieLens 1M Test Sets

The detection results of the three methods (i.e. Single SVM, Voting SVM and Meta SVM) on the MovieLens 1M test sets are shown in Table Ⅲ, Table Ⅳ, Figure 4 and Figure 5.

TABLE Ⅲ.

RECALL ON MOVIELENS 1M TEST SETS FOR VARIOUS ATTACK MODELS AT FILLER SIZE=3% ACROSS VARIOUS ATTACK SIZES

Attack Model	Random Attack				Bandwagon Attack				Average Attack			
Attack Size	1%	2%	5%	10%	1%	2%	5%	10%	1%	2%	5%	10%
Single SVM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Voting SVM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Meta SVM	0.97	0.98	0.98	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99

TABLE Ⅳ.

RECALL ON MOVIELENS 1M TEST SETS FOR VARIOUS ATTACK MODELS AT FILLER SIZE=5% ACROSS VARIOUS ATTACK SIZES

Attack Model	Random Attack				Bandwagon Attack				Average Attack			
Attack Size	1%	2%	5%	10%	1%	2%	5%	10%	1%	2%	5%	10%
Single SVM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Voting SVM	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Meta SVM	0.98	0.97	0.97	0.95	0.95	0.98	1.00	1.00	0.99	1.00	1.00	0.99

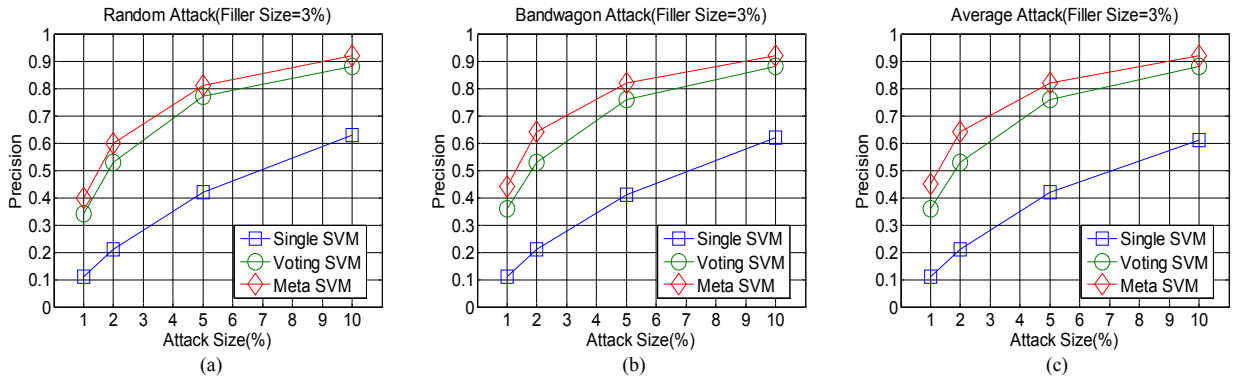


Figure 4. Precision on MovieLens 1M test sets for various attack models at filler size=3% across various attack sizes

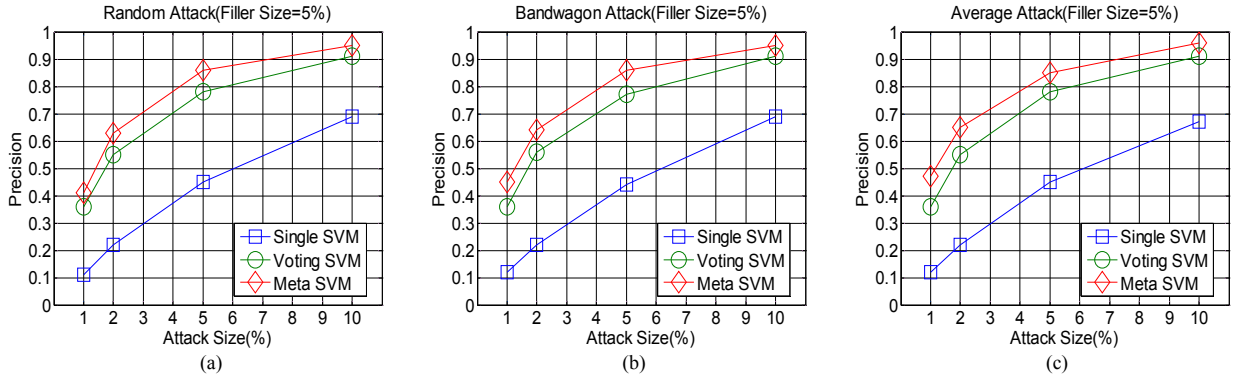


Figure 5. Precision on MovieLens 1M test sets for various attack models at filler size=5% across various attack sizes

As shown in Table Ⅲ, Table Ⅳ, Figure 4 and Figure 5, the recalls and precisions of Meta SVM on MovieLens 1M test sets, whose scale are larger than the scale of MovieLens 100K test sets, still keep at a high level.

In Table Ⅲ and Table Ⅳ, most recalls of Meta SVM are 0.99 and the minimum value has reached 0.95. These recalls are slightly lower than the recalls of Single SVM and Voting SVM. However, we believe that Meta SVM is more practical than the other two detection methods. Blew, we show the explanations for this viewpoint. In a recommender system, a group of attack profiles is particularly harmful against the recommendation results while the attack effects of several attack profiles is very limited [7][10]. In the worst case, the recall of Meta SVM only drops by 0.05 percentage points compared to the recall of Single SVM. Meta SVM can detect most of the attack profiles under the condition of holding high precisions while Single SVM and Voting SVM improve their recalls through increasing the misclassifications.

In Figure 4 and Figure 5, the precisions of Meta SVM increase with increasing of attack size, take Figure 4(a) for example, when the attack sizes are 1%, 2%, 5%, 10%, the precisions of Meta SVM are 0.4, 0.6, 0.81 and 0.92. These trends are also shown in Figure 2 and Figure 3. This is not hard to explain, with increasing of attack size, more attack profiles are detected while the number of misclassified genuine profiles keeps the same. These illustrate that Meta SVM has a good stability when detecting the attacks at large attack sizes.

The precision curves of Voting SVM are closer to the curves of Meta SVM in Figure 4 and Figure 5 than the curves in Figure 2 and Figure 3. This illustrates that the classification performance of Voting SVM is improved on the large-scale dataset. However, the precisions of Meta SVM are always higher than the precisions of Single SVM and Voting SVM. At 1% attack size, the average precision of Meta SVM increases by 3 percentage points and 0.2 percentage points compared to

the average precision of Single SVM and Voting SVM respectively. At 10% attack size, while the average precisions of Single SVM and Voting SVM are 0.58 and 0.85 respectively, the average precision of Meta SVM reaches 0.94 higher than the average precision 0.91 in Figure 2 and Figure 3. These results illustrate that Meta SVM can effectively detect the attack profiles not only on a small-scale dataset but also on a large-scale dataset with a high precision.

V. Conclusions and Future Work

The detection of profile injection attacks is an important research area in the recommender system. We have made some beneficial explorations and attempts in this area. We propose an algorithm to create the diverse base-level training sets through flexible combinations of various attack types. Using these sets, we construct the diverse base-level classifiers to reduce the correlations of the misclassifications. Through relearning the results of the base-level classifiers, a meta-classifier is constructed. Based on base-level and meta-level classifiers, we propose a meta-learning-based detection algorithm which could effectively detect profile injection attacks. The experimental results on the different-scale datasets prove that the proposed approach can not only hold a high recall, but also effectively improve the precision.

In the future, we will study the more appropriate meta-learning strategies to improve the detection performance for profile injection attacks in collaborative recommender systems.

Acknowledgment

This work is partially supported by the Natural Science Foundation of Hebei Province, China (F2011203219) and the Special Fund for Fast Sharing of Science Paper in Net Era by CSTD of Ministry of Education, China (20101333110013).

References

[1] I. Bartolini, Z. Zhang, and D. Papadias, “Collaborative filtering with personalized skylines,” IEEE Transactions on Knowledge and Data Engineering, vol.23(2), pp.190-203, February 2011, doi:10.1109/TKDE.2010.86.

[2] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, “Detecting profile injection attacks in collaborative recommender systems,” In Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE’06), June 2006, doi:10.1109/CEC-EEE.2006.34.

[3] S. K. Lam, and J. Riedl, “Shilling recommender systems for fun and profit,” In Proceedings of the 13th international conference on World Wide Web (WWW'04), pp.393-402, May 2004, doi:10.1145/988672.988726.

[4] R. Burke, B. Mobasher, and R. Bhaumik, “Limited knowledge shilling attacks in collaborative filtering systems,” In Proceedings of the 3rd IJCAI Workshop in Intelligent Techniques for Personalization (ITWP’05), August 2005, doi:10.1.1.135.2204.

[5] R. Burke, B. Mobasher, R. Bhaumik, and C. Williams, “Segment-Based injection attacks against collaborative filtering recommender systems,” In Proceedings of the International Conference on Data Mining (ICDM 2005), pp. 577-580, November 2005, doi:10.1109/ICDM.2005.127.

[6] M. O’mahony, N. Hurley, N. Kushmerick, and G. Silvestre, “Collaborative recommendation: a robustness analysis,” ACM Transactions on Internet Technology, vol.4(4), pp.344-377, November 2004, doi:10.1145/1031114.1031116.

[7] B. Mehta, and W. Nejdl, “Attack resistant collaborative filtering,” In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR’08), pp.75-82, July 2008, doi:10.1145/1390334.1390350.

[8] P.-A. Chirita, W. Nejdl, and C. Zamfir, “Preventing shilling attacks in online recommender systems,” In Proceedings of the 7th annual ACM international workshop on Web information and data management (WIDM ’05), pp.67-74, November 2005, doi:10.1145/1097047.1097061.

[9] X.-F. Su, H.-J. Zeng, and Z. Chen, “Finding group shilling in recommendation system,” Special interest tracks and posters of the 14th international conference on World Wide Web (WWW'05), pp.960-961, May 2005, doi:10.1145/1062745.1062818.

[10] B. Mehta, T. Hofmann, and P. Fankhauser, “Lies and propaganda: detecting spam users in collaborative filtering,” In Proceedings of the 12th international conference on Intelligent user interfaces (IUI’07), pp.14-21, January 2007, doi:10.1145/1216295.1216307.

[11] B. Mehta, “Unsupervised shilling detection for collaborative filtering,” In Proceedings of the 22nd national conference on Artificial intelligence (AAAI'07), pp.1402-1407, July 2007.

[12] B. Mehta, and W. Nejdl, “Unsupervised strategies for shilling detection and robust collaborative filtering,” User Modeling and User-Adapted Interaction, vol.19(1-2), pp.65-79, 2009, doi:10.1007/s11257-008-9050-4.

[13] K. Bryan, M. O’Mahony, and P. Cunningham, “Unsupervised retrieval of attack profiles in collaborative recommender systems,” In Proceedings of the 2008 ACM conference on Recommender systems (RecSys’08), pp.155-162, October 2008, doi:10.1145/1454008.1454034.

[14] N. Hurley, Z. Cheng, and M. Zhang, “Statistical attack detection,” In Proceedings of the third ACM conference on Recommender systems (RecSys’09), pp.149-156, October 2009, doi:10.1145/1639714.1639740.

[15] F. He, X. Wang, and B. Liu, “Attack detection by rough set theory in recommendation system,” IEEE International Conference on Granular Computing, pp.692-695, August 2010, doi:10.1109/GrC.2010.130.

[16] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, “Classification features for attack detection in collaborative recommender systems,” In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD’06), pp.542-547, August 2006, doi:10.1145/1150402.1150465.

[17] C. A.Williams, B. Mobasher, and R. Burke, “Defending recommender systems: detection of profile injection attacks,” Service Oriented Computing and Applications, vol.1(3), pp.157-170, 2007, doi:10.1007/s11761-007-0013-0.

[18] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, “Towards trustworthy recommender systems: an analysis of attack models and algorithm robustness,” ACM

Transactions on Internet Technology, vol.7(4), October 2007, doi:10.1145/1278366.1278372.
[19] Z. Fuzhi, J. Dongyan, C. Jinbo, "An improved PCA attack detection algorithm based on normal cloud model," Journal of Computational Information Systems, vol.6(6), pp. 1959-1966, 2010.
[20] Z. Fuzhi; J. Dongyan, C. Jinbo, "A user profile injection attack detection algorithm based on normal cloud model and PCA," ICIC Express Letters, vol.5(4A), pp. 925-930, 2011.
[21] X. Yuchen, L. Qiang; Z. Fuzhi, "User profile attack anomaly detection algorithm based on time series analysis," Journal of Information and Computational Science, vol.7(11), pp. 2201-2206, 2010.
[22] R. Bhaumik, C. Williams, B. Mobasher, R. Burke, "Securing collaborative filtering against malicious attacks through anomaly detection," In Proceedings of the 4th workshop on intelligent techniques for web personalization (ITWP'06), July 2006.
[23] Z. Lan, J. Gu, Z. Zheng, R. Thakur, and S. Coghlan, "A study of dynamic meta-learning for failure prediction in large-scale systems," Journal of Parallel and Distributed Computing archive, vol.70(6), pp.630-643, June 2010, doi:10.1109/ICPP.2008.17.
[24] D. H. Wolpert, "Stacked generalization," Neural Networks, vol.5(2), pp.241-259, 1992, doi:10.1.1.56.1533.
[25] P. Kordík, J. Koutník, J. Drchal, O. Kovářík, M. Čepek, and M. Šnorek, "Meta-learning approach to neural network optimization," Neural Networks, vol.23(4), pp.568-582, May 2010, doi:10.1016/j.neunet.2010.02.003.
[26] M. Wasikowski, X.-W. Chen, "Combating the small sample class imbalance problem using feature selection," IEEE Transactions on Knowledge and Data Engineering, vol.22(10), pp.1388-1400, October 2010, doi:10.1109/TKDE.2009.187.
[27] Y. Fu, H. Yan, J. Li, and R. Xiang, "Robust facial features localization on rotation arbitrary multi-view face in complex background," Journal of Computers, vol.6(2), pp.337-342, February 2011, doi:10.4304/jcp.6.2.337-342.
[28] C.-F. Tsai, Y.-C. Linc, D. C.Yen, and Y.-M. Chen, "Predicting stock returns by classifier ensembles," Applied Soft Computing, vol.11(2), pp.2452-2459, 2011, doi:10.1016/j.asoc.2010.10.001.

Fuzhi Zhang was born in 1964. Currently, he is a professor and PhD supervisor in School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. His research interests include intelligent information processing, network and information security, and service-oriented computing.

Quanqiang Zhou was born in 1985. Currently, he is a PhD student in School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. His research interests include intelligent information processing and personalization.

The Design of SMS Based Heterogeneous Mobile Botnet

Guining Geng, Guoai Xu, Miao Zhang and Yanhui Guo
Information Security Center,
Beijing University of Posts and Telecommunications, Beijing, China
gengguining@163.com, {xga, zhangmiao, yhguo }@bupt.edu.cn
Guang Yang
China Information Technology Security Evaluation Center, Beijing, China
sunwina@126.com
Wei Cui
Information Center of Ministry of Science and Technology of the People's Republic of China
cuiw@most.cn

***Abstract*—Botnets have become one of the most serious security threats to the traditional Internet world. Although the mobile botnets have not yet caused major outbreaks worldwide in cellular network, but most of the traditional botnet experience can be transferred to mobile botnet on mobile devices, so mobile botnet may evolve faster since techniques are already explored. From the theoretical work of some researchers and the reports of security companies, we can see that the mobile botnet attacks and trends are quite real. In this paper, we proposed a SMS based heterogeneous mobile botnet, and shown how SMS based C&C channel structure can be exploited by mobile botnets. At last, we give the analysis of connectivity, security and robustness evaluation of our model.**

***Index Terms*—heterogeneous, mobile botnet, SMS, C&C, robustness**

I. INTRODUCTION

Traditional botnets have become one of the most serious security threats to the Internet. The word "bot" means that those victims controlled by attacker, and it derives from the word "robot". A bot master can control a large scale of bots at different locations to initiate attack, and due to the complexity of the internet, it can be hardly trace back, and lots of researchers have done remarkable studies about the traditional botnet, such as botnet threats[1], botnet model[2], control strategies[3] and botnet detection[4].

Compared with the evolution of traditional botnet on the Internet world, mobile botnet are 7-8 years[5] behind. Although the mobile botnets have not yet caused major outbreaks worldwide in cellular network, but most of the traditional botnet experience can be transferred to mobile botnet. Norman[5] predicts that malware on mobile device(smart phones) will evolve faster since techniques are already explored.

The early malware can only perform one or two tasks, like Cabir. Cabir was detected in 2004, it is the earliest mobile botnet we ever known.

But in 2009, the situation has changed dramatically. Mobile bots can connect back to a malicious bot server and transfer valuable information of the infected device, like SymbOS.Exy.C[6], Ikee.B[7,8]. According to the report of Symantec[6] on 13 July 2009, SymbOS.Exy.C may the first bot on Symbian OS. It is a worm similar to other worms that made for Symbian OS, but the difference is that the bot node tries to contact a malicious bot server and transfer valuable information, such as the phone types, International Mobile Equipment Identity (IMEI) and International Mobile Subscriber Identity (IMSI)[9]. Symantec mentions that this may be the first true botnet occurred on a cellular mobile device.

BBOS_ZITMO.B[10] was detected in 2011. It receives commands via SMS, steal users information by forwarding SMS messages to a set/predefined admin phone number, monitors incoming calls and SMS. The most important is that it also has a stealth mechanism that prevents being seen as an installed app.

Figure1 illustrates the simplified typical infection cycle of an SMS based mobile botnet. There are mainly 4 steps in the botnet operation [11]. Vulnerable smart phones could be first infected by bots in the original botnet. The existed techniques (smart phone OS vulnerabilities, Trojan horses, worms, etc) were used during the infection period. After the infection, the infected node connects to the bot server to join in the origin botner. All the bot servers are organized as the C&C network and controlled by bot master. Bot master use the C&C network to issue commands, and control the whole botnet. According to our analysis, the bots in the same tier or sub network could launch the corresponding attack after receiving the command. Authorization is achieved via a channel password.

doi:10.4304/jcp.7.1.235-243

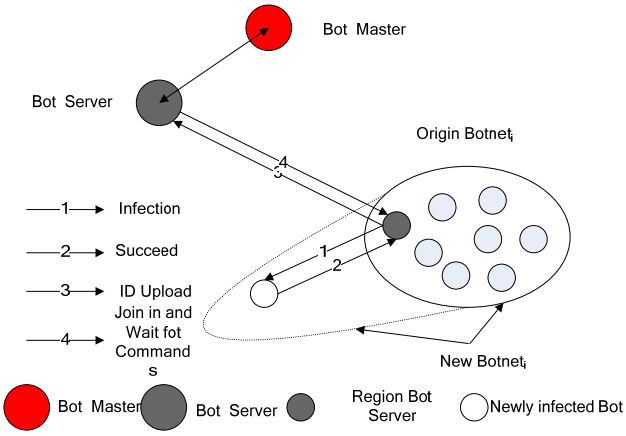


Figure 1 Typical infection cycle

Consider the potential threats of the mobile botnet to the cellular network. Researchers have done some remarkable research about the attacks on cellular network, such as DoS attack [12], C&C channel study[13], paging channel overloads [14].

In this paper, we proposed the SMS based heterogeneous mobile botnet. The C&C channel is SMS based heterogeneous multi-tree structured network. That is, all C&C commands are transfered via SMS messages since SMS is available to almost mobile phones. We use heterogeneous multi-tree topology to enhance the scalability and robustness of the botnet. We also made all the bot lists and some of the important commands encrypted.

Our research has the following main contributions:

- We proposed an improved SMS based heterogeneous mobile botnet. The heterogeneous bot nodes and networks structure have made the mobile botnet more efficient to communicate and more secure.
- Contrast with the traditional botnet of Internet world, we give the definition of mobile botnet and illustrate the characteristics of the mobile botnet.
- We introduced node degree threshold $\langle k \rangle$ and mobile botnet height H to improve the security of the C&C channel. The improved C&C channel raises the bar for the countermeasure of mobile botnet community.
- Through the design of the eviction and replacement mechanism of failed or recovered bot server node, the robustness of C&C channel was enhanced.

The remainder of the paper is organized as follows. Section II shows the characteristics of cellular bonnets. Section III illustrates the mobile botnet attacks. Section IV proposed our SMS based heterogeneous cellular mobile botnet. Section V discussed our model from the aspect of connectivity, security and robustness. The paper concludes and future work with Section VI.

II. THE CHARACTERISTICS OF MOBILE BOTNETS

In this paper we define the mobile botnet as follows:

Definition 1 mobile botnet: Consists of a network with compromised smart phones, controlled by attacker (“bot master”) thorough a command and control(“C&C”) network for malicious purposes.

The mobile botnets are different from the traditional botnets of Internet world. Since the mobile botnets are mostly communicated between mobile devices (smart phones), it has the following characteristics.

- Limited by the power resource. The mobile devices such as smart phones are different from PC, its run time is limited due to the use of battery.
- The communication costs problems. The communications of mobile botnets will consume the limited power resource, network traffic and especially the phone charge. The communications of the mobile botnets will lead to the cost of the owner, and a significant rise of the phone charge will result in the investigation of the cause and thus may lead to the exposure of the culluar bot. The SMS based mobile botnets, depending on the type of mobile phone contract, SMS messages can be completely free or charging very few money.
- The connectivity changes constantly, even unstable. The connectivity may be both affected by physical environment or personal factors. It can be affected by the networks around the mobile phone owner, the action of the mobile phone owner, such as the user is in the tunnel or turn off the mobile phone during the bed time. As shown in Table 1, the connectivity of bots can be changed constantly during the daily life.

Table 1
THE CONNECTIVITY AND TIME INTERVAL

Connectivity	*Time Intervals*
WiFi	Morning (at home)
GSM/EDGE/3G	Day time (at work/school)
WiFi	Day time (at Starbucks)
No signal	Day time (at wild)
WiFi	Evening (at home)
Turn off the mobile phone	Night (bed time)

- Lack of IP address. The lack of IP address may cause the problem of indirect connect. Due to the lack of IP address, most mobile phones are using NAT gateway and thus the devices are not directly reachable, so the traditional P2P based C&C network may not suit for mobile botnet.
- The diversity of operating system of smart phone. The design of mobile botnet has to consider the diversity of the OS platform of smart phone.
- Mobile botnets are hard to be detected. Evidences indicate that, mobile botnets are becoming more and more sophisticated[10]. The valuable messages can be forwarded by SMS messages to a predefined server device, and the SMS messages are deleted immediately after they were forwarded by mobile botnets, so it is hard to be detected. It also has a stealth mechanism that prevents being detected as an installed app..

III. MOBILE BOTNET ATTACK

A. Information Leakage

One of the main targets of the mobile botnet is to retrieve sensitive information from the victims. The mobile bot can quickly scanning the host node for significant corporate or financial information, such as usernames and passwords, address list and text messages.

B. DoS Attack

Because most of the functionality of cellular network rely on the availability and proper functioning of HLRs(Home Location Register), so the DoS attack could block the legitimated users of a local cellular network from sending or receiving text messages and calls[9,12,15,17].

In the practical circumstances, a bot master of a mobile botnet could control the compromised mobile phones to overwhelm a specific HLR with a large volume of traffic. Through the DoS attack, it will affect all the legitimated users who rely on the same HLR, their requests will be dropped.

By overloading (red arrow) the HLR of a local central component of the cellular network, the network will unable to server legitimated traffic (blue arrow) of very large geographic regions. Figure 2 shows how a cellular network DoS attack occurs.

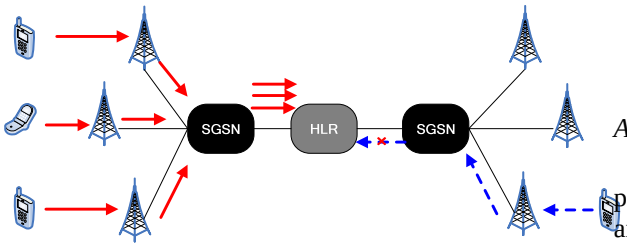


Figure 2 A cellular network DoS attack

C. Charge loss

There exist some service which the smart phones can give money to charity organizations[17]. If the mart phone called or sent a text message to the specific service number, then the subscriber will pays a preset amount of money. The bot master can also creates its own service number and programs all the bots to call or sent a text message to the specific service number. Of course, the price should be low, so the subscribers would not notice and be suspicious about the extra charges.

IV. SMS BASED HETEROGENEOUS MOBILE BOTNET DESIGN

As we known, the three main components of botnet are vectors, C&C channel and network topology. The vector solved the problem of how to spread bot code, and can be used in propagation period; C&C channel is used to issue command and control messages; the network topology of botnet is to manage the botnet and enhance the security and stealth of botnets.

Considering the problems that have been encountered during the design period of traditional botnets and mobile botnets, we think the bot master should consider the following challenges:

1. How to transmit command and control messages and control traffic flow.
2. How to prevent the proposed mobile botnet from being detected by defenders.
3. How to monitor the status of bot nodes.
4. How to enhance the robustness of mobile botnet even some bot nodes have been removed by the defenders.

In this section, we will propose the SMS based heterogeneous mobile botnet, and briefly overview our mobile botnet and all type of nodes and then focus on the propogation, network topology, command and control mechanism, node replacement mechanism of our mobile botnet. The designed mobile botnet model is shown in Figure 3.

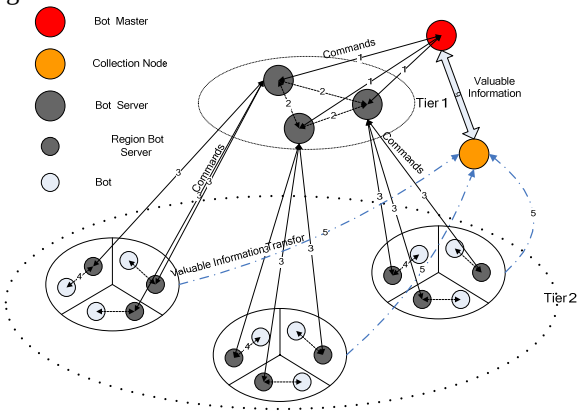


Figure 3 The simplified SMS-based heterogeneous mobile botnet

A. Node Types

The infected node can be classified by the difference of performance, battery resource, connectivity and networks around it, so they can be divided into several types of bot nodes. As is illustrated in Figure 3, our mobile botnet is made up of bot master, collection node, bot servers, region bot servers and a certain amount of bot nodes. The topology of our mobile botnet was designed multi-tree structured from top to down, but the topology of the same tier nodes such as bot server tier was designed P2P structured. So, the heterogeneous exist both in bot nodes types and network topology.

Bot master node: Bot master controls all the nodes of the mobile botnet. The bot master has direct contact with bot servers. It has the bot list of all the botnet, and it stores all the node's information, like user name, phone number, international mobile subscriber identity (IMSI), International Mobile Equipment Identity(IMEI). Unless an emergency situation, bot master will not communicate with bot node directly.

Collection node: It receives the valuable information from all the nodes of the botnet. The stored information can be fetched by bot master. Before it receives the information, the node that sends information must be authenticated.

Bot server node: Bot Server node is one of the key nodes of our model. It has two main functions, search and forward, it has k' direct links with region bot servers, and each of them controls a sub network. It has the bot list that includes all the nodes it communicated with, and the bot list is encrypted.

Region bot server node: Region bot server node is another key node of our model. It is both boy server and bot node. It receives commands from bot server and forwards to the bots of its sub network. It executes the commands and transfer valuable information to the collection node. Also its bot list was encrypted.

Bot node: Bot node is the leaf node of ourmobile botnet. It receives commands from region bot server, and executes the commands.

The comparison of communication and bot list between the five types of nodes that we have stated is shown in table2.

Table 2
THE COMMUNICATION OF MODEL NODES

Node type	Bot master		Collection node		Bot server		Region bot server		Bot	
	Commu.	Node list	Commu.	Node list	Commu.	Node list	Commu.	Node list	Commu.	Node list
Bot master		yes	yes	yes	yes	yes				
Collection node	yes	yes					yes	yes	yes	
Bot server	yes	yes			yes	yes	yes	yes		
Region bot server		yes	yes	yes	yes	yes			yes	
Bots		yes	yes	yes			yes	yes	yes	yes

B. Propogation

We divide the propagation phase into three steps.

In the first step, the bot master exploit the operating system and configuration vulnerabilities to compromise the mobile devices, install the bot software and collecting valuable information. After the bot software was installed, we assume the bot node has a stealth mechanism that prevents being seen as an installed app, and the messages that used to collect valuable information were deleted.

In the second step, the bot master choose bot server nodes from the compromised nodes as the first tier nodes according to the following conditions, the energy resource, the region where it belongs to and the connectivity. The rest of the nodes that are not chosen as the bot servers, assigned to the bot servers according the regions, as the second tier nodes.

The third step, the nodes of the existed botnets continued infecting mobile devices under the control of the bot master. According the expansion of the mobile botnet, the number of tier will grow.

C. Network Topology

Definition 2 Botnet degree: The degree of our mobile botnet is *K*. It is the maximum value of $\langle k_i \rangle, i \in N$.

Definition 3 Botnet height: The height of our mobile botnet is *H*. It is the number of tiers of the model.

Definition 4 A**tomic network:** The atomic network is composed of region bot node and $k^{'}$ bot nodes, it is the smallest bot network that was control by region bot server at the lowest tier.

Definition 5 $k^{'}$**:** we define $k^{'}$ is the number of lower tier nodes that communicate with higher tier node n_i.

In general, our proposed mobile botnet is made up of 1 bot master, $k^{'}$ bot servers, $k^{'}$ region bot servers, and $k^{'}$ sub networks.

The maximum degree of bot server n_i is $\langle k \rangle = 2k^{'}$, in Figure 3, $k^{'}$ =3, $\langle k \rangle$ =6. For mobile botnets with large scale bot nodes, it makes sense to limit the degree of botnet *K* under a threshold. So, unless the bot nodes are high capacity server, bot masters should keep $\langle k \rangle$ small[14].

As we said, the topology of our model is heterogeneous multi-tree structured. From top to down, the model is a multi-tree structure network. The bot server tier and region bot server tiers are the P2P structured networks. The degree of the model is *K*, the height of the model is *H*. The degree of node n_i is $\langle k_i \rangle$, *K* is the maximum number of the degree of all the nodes, $K = \max(\langle k_1 \rangle, \langle k_2 \rangle, ..., \langle k_N \rangle)$. Our botnet(see Figure 3) can be divided into *H*-1 tiers and $k^{'}$ subnets. The lowest tier network contains $k^{'}$ ($k^{'}$ =3) subnet, and each subnet is composed of $k^{'}$ P2P structured atomic networks, as is shown in Figure 4.

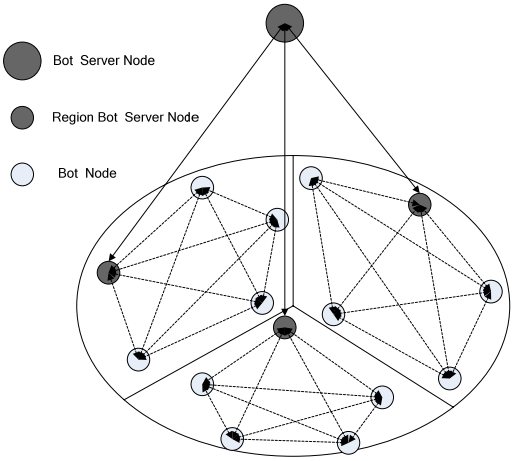


Figure 4 The simplified topology of the subnet of mobile botnet

The topology between bot server nodes is P2P structured network (tier 1in Figure 3), the distances between them are all 1 hop. Also the topology between region bot server nodes and bot nodes are P2P structured

network. This will enhance the communication efficiency between the mentioned bot nodes,.

D. Command and Control Network

In our mobile botnet model, we use SMS as the C&C channel. Most of the traditional botnets are using centralized IRC, HTTP protocol and P2P based C&C channel, and all of these channels are IP-based C&C delivery. Unlike the PC world, even with the edge or 3G networks, the smart mobile devices (smart phones) still can not establish stable IP based connections with each other. Given this limitation, in our work we use SMS as our un-centralized and non-ip-based C&C channel.

The advantages of SMS based C&C channel can be listed as follows.

First, it is popular. Most of mobile phone subscribers are sending and receiving text messages to each other.

Second, it is easy to use. Malicious content can be hidden in message, the command and control messages can be disguised as spam-looking messages.

Third, it has high connectivity. Even a phone has signal problem or power off, the SMS messages that send to it will eventually arrive to it, because the SMS messages can be stored in a service center and delivered once the signal becomes available or turned back on[19].

Our mobile botnet can be considered as “PUSH” based botnet. The “PUSH” based botnet deliver commands form bot server to bot nodes whenever the bot master wants, and we call this process “PUSH”. On the contrary, the traditional botnet [20] and [21] are the “PULL” based botnets, the bot nodes of them periodically communicate with bot server to get commands. But the “PULL” based botnets have some drawbacks, such as generate additional information, have unexpected latency in command delivery. G. Gu et al [18] has designed an effective detection schemes for it. All of the above statements made the “PULL” based botnets unattractive in practice. The “PUSH” based delivery does not generate additional information, and have little latency in command delivery, in this paper, we design our model as “PUSH” based botnet.

As is illustrated in Figure 3, The basic design idea for our SMS-based mobile botnet is to use a heterogeneous multi-tree network as the Command and Control(C&C) networks, the network is like a commands channel, all the nodes of network(black nodes) are served as bot server. The C&C network of our proposed model is composed of bot master, bot server and region bot server.

Like Kademlia[22], the absence of authentication mechanism means that anyone can insert values to the commands, the defender may use this launch index poisoning attacks and the C&C may be disrupted. So before communication the nodes should be authenticated, and the important command should be encrypted. For the efficiency of C&C, only critical commands issued by the bot master that order bots to execute malicious tasks such as “transmit valuable information” are encrypted, while commands for P2P communication purposes such as “search node” are only disguised without encryption[19].

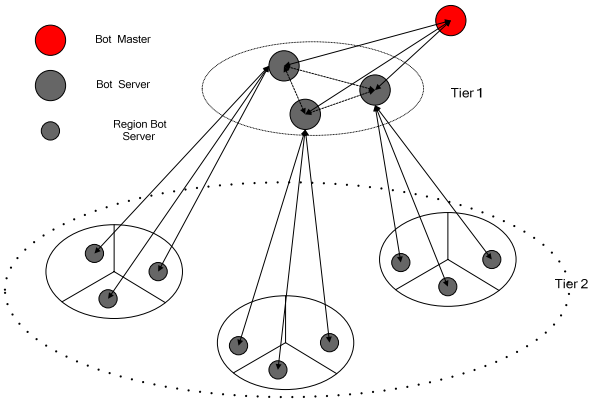


Figure 5 The C&C network

According to the bot list of it, bot server node performs two main function, search and forward. Some of the important commands are encrypted, and the encrypted commands are pushed from top tier nodes to lower tier nodes, from the bot master to bot server node, to region bot server node and bot node respectively. All of the bot lists are encrypted, once the key node was captured, the defender will not get the other nodes that communicate with it. And, also it cannot trace the bot master.

E . Node repalcement mechanism

At some special circumstances, bot node type may need to transfered from each other. Like one of the bot server node is failed or recovered.

The bot server nodes and region bot server nodes are the key nodes of our model, so they must be replaced by other nodes if they are out of function, such as failed or recovered. As is shown in Figure 6, at some circumstances, bot node type may need to transfer from each other. Since bot master has the bot list of the failed key node, by sending a message that containing the failed node’s bot list and communication keys to the new key node and that will make it reconnected to the bots.

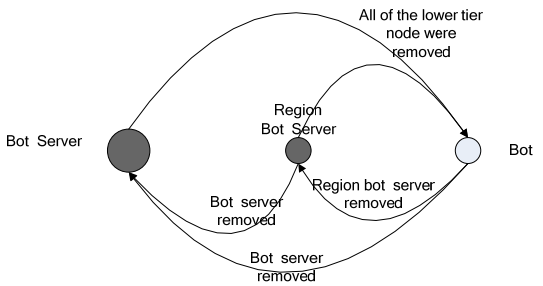


Figure 6 The transfer between three type of nodes

1) Node Eviction

Assumption1: we assume that the failed or recovered nodes can be detected by bot master.

Assumption2: we assume that all bot node have a self destroy mechanism to delete the bot list that stored in it.

Table 3
THE LIST OF NOTATIONS

Notation	Description
K_{BB}	Keys between bot server and bot master
K_{BS}	Keys between bot servers
K_{BR}	Keys between bot server and region bot server

First, the bot master detect the failed or recovered node n_i, then according to phone number of n_i that bot master stored in it to refresh the according keys, such as K_{BB}, K_{BS}, K_{BR}. Second, the bot master sends $k'-1$ messages that contain the new keys K_{BS} to all the other bot servers, and send k' messages that contain K_{BR} to the region bot servers, which n_i communicate with. Third, the phone number of n_i was deleted from the bot list of bot master, bot server and region bot server.

After the eviction period, the failed or recovered node n_i can not communicate with the bot nodes that we have mentioned, and also the messages that send to them were discarded automatically. The eviction procedure is illustrated in Figure 7.

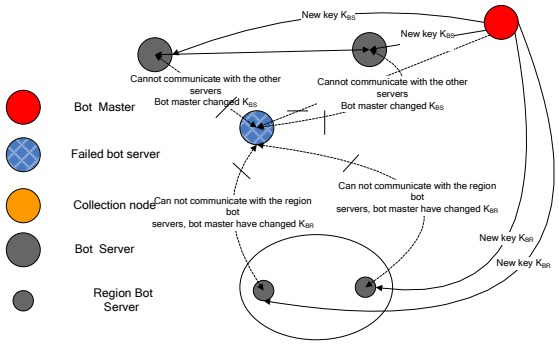


Figure 7 The eviction of failed or recovered bot server node

2) Node Replacement

At the replacement period, first, bot master choose a relatively powerful node from low tier bot nodes or region bot server as the new bot server according to standers that we have stated in *propagation* section. Second, the bot master send 1 message that contains the new keys K_{BB}, K_{BS}, K_{BR} and bot list to the new assigned bot server. Third, the new bot serner need to send $k'-1$ messages to establish communication with the rest bot servers and k' messages to establish communication with the region bot servers that were communicated with the captured bot server, so the total number of messages during the replacement period is $2k'-1$. The replace procedure is illustrated in Figure8.

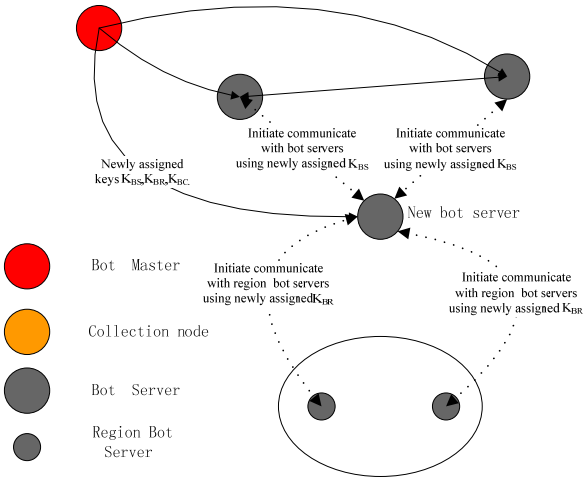


Figure 8 The replacement of bot server node

V. EVALUATION

In this section, we would like to analyze our proposed mobile botnet from the aspect of connectivity, security, and robustness.

A. Connectivity

In this section, we will discuss the connectivity of our heterogeneous botnet with different scale of node, and compare the connectivity with the P2P structured botnet.

We will look at the 200-node botnet first and then the 1000-node and 2000-node botnet. In the 200-node botnet, when k'=4 and k'=5, all the nodes in the botnet can be reached in 4 hops, and when k'=3, all the nodes in the botnet can be reached in 5 hops, as is shown in Figure 8. In the 1000-node botnet, as is illustrated in Figure 9, all the nodes can be reached in 5 hops when k'=4 and k'=5, and when k'=3, all the nodes can be reached in 6 hops. Figure 10 shows the 2000-node botnet, all the nodes will be reached in 5, 6, 7 hops respectively.

From Figure 9, 10, 11, we can see that the larger of k', the less hops mobile botnet needs to reache all the bot node of it, when the number of hops larger than 3, the connectivity will rise dramatically.

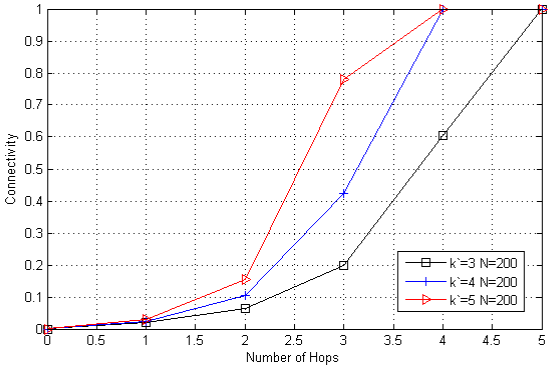


Figure 9 The relationship between connectivity and k', when N=200

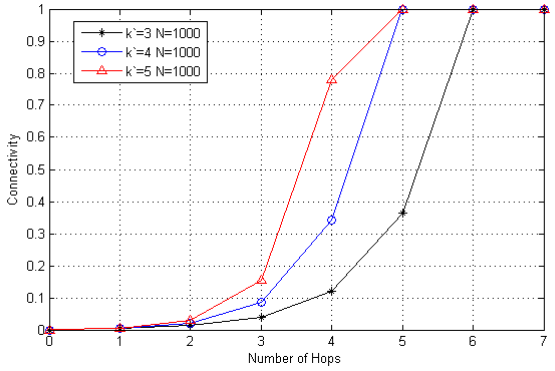


Figure 10 The relationship between connectivity and k', when N=1000

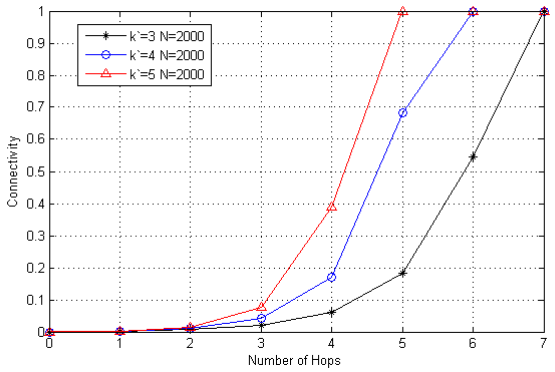
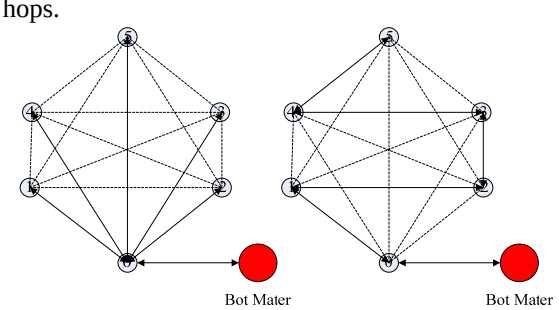


Figure 11 The relationship between connectivity and k', when N=2000

We compare our proposed botnet with the P2P structured botnet. In the P2P structured botnet, the average path length between any two nodes can be 1 hop to N hops.



(a)The shortest path length (b)The longest path length

Figure 12 the path length of P2P structured Botnet

As is illustrated in Figure 11(a), in the P2P structured botnet, the commands from Bot_0 can be directly forward to Bot_1, Bot_2, Bot_3, Bot_4, Bot_5. So the mean path length is

$$PL_{P2P\,structured\_1} = \frac{\sum_{i=1}^{N-1} l_i}{N-1} = \frac{\sum_{i=1}^{N-1} 1}{N-1} = 1 \qquad (1)$$

In Figure 8(b), the commands from Bot_0 can finally forward to Bot_5, through Bot_1, Bot_2, Bot_3 and Bot_4. So the mean path length is

$$PL_{P2P\,structured\_2} = \frac{\sum_{i=1}^{N-1} l_i}{N-1} = \frac{\sum_{i=1}^{N-1} i}{N-1} = \frac{N}{2} \qquad (2)$$

According to Formula (1) and (2), the mean path length of the P2P structured botnet is

$$Mean_{P2P\,structured} = \frac{1+\frac{N}{2}}{2} = \frac{2+N}{4} \qquad (3)$$

So the mean connection hops between two nodes of the P2P structured botnet is $\frac{2+N}{4}$. When N =6, the mean connection hops is 2, it has nice network connectivity, but when N=200 and 2000, the mean connection hops are approximately 50 and 500, respectively, and it larger than our proposed botnet.

From the comparison, we can see that our proposed mobile botnet have high connectivity and it can be adjusted easily by adjust the height and scale of mobile botnet. It has high flexibility and scalability.

B. Security

As we described in section IV, the height of the model is H, k' is the number of lower tier nodes that communicate with heighter tier node. As is illustrated in Figure 5, the maximum number of nodes at the lowest tier is $(k')^H$. Then, the maximum number of our proposed botnet model is

$$N = \sum_{i=0}^{H-2} (k')^i + (k')^H \qquad (1)$$

The larger of the H, the less community load of the key node. But large H will lead to long cummunication path and low connectivity. The relationship between model degree K and node degree $\langle k_i \rangle$ is

$$K = \max(\langle k_1 \rangle, \langle k_2 \rangle, ..., \langle k_N \rangle) \qquad (2)$$

$$\langle k \rangle \le 2k' \qquad (3)$$

As we can see form Formula (2) and (3), K increase with the increase of k'. During the C&C period, the key nodes have to forward at least k' messages at one time, but the large k' means that there is large scale of abnormal data traffic, and that will jeopardize the safety of key nodes. The relationships of k', N, and H are shown in Figure 9. When k' =3 and the value of H are 2, 3, 4, 5 respectively, the proposed botnet will have better connectivity but have lower capacity. When the value of k' is 5, there is a better leverage between connectivity and N. In this paper, consider the tradeoff between k' and N, we set our mobile botnet H=5, $k' \le 5$, so $\langle k \rangle \le 10$, $K \le 10$, and the scale of the botnet can up to 3281 bot nodes.

By the tradeoff between k' and H, we can efficiently control the amount of traffic flow that travels between bot nodes, and the communications between bot nodes will not catch the attention of defender and then protect the mobile botnet, so our mobile botnet can have high security.

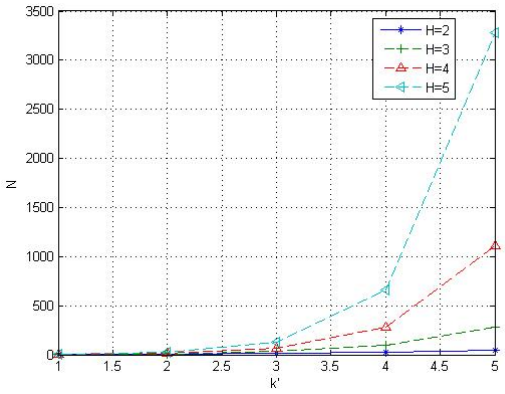


Figure 13 The relationship of *k'*, *N* and *H*

C. Robustness

As we discussed in section II, Battery consumption plays a very important role in the cellular world. So it is a challenge for the mobile bot not to drain the battery significantly, otherwise, it will jeopardize the security of the bot nodes. High CPU load and heavy forward usage are the main causes of the battery drain of bot node.

Based on our study, radio usage cost more energy, so we are more concerned about the forward usage. Hence our mobile botnet has "PUSH" based delivery mechanism, the commands from the bot master are direct transferred by bot servers to bot nodes, and no additional messages are generated.

Before C&C channel repair, the bot master need to check if the communication multi-tree has broken at some key nodes, all it has to do is broadcast ping and every node need to answer it. Obviously, the check of the communication multi-tree is important, and need to perform periodically. The check may also perform at some sub networks at one time so that it will not cause the attention of the defender.

As is shown in Figure 14, if the height of botnet is 5, the maximum degree of 200-botnet is 3, the maximum degree of 200-botnet is 5, the eviction needed messages of 200-botnet are 5, and the eviction needed messages of 2000-botnet are 9. As we have seen that the number of eviction needed messages are acceptable and the eviction of recovered bot node of our botnet is feasible and can save the limited energy resource.

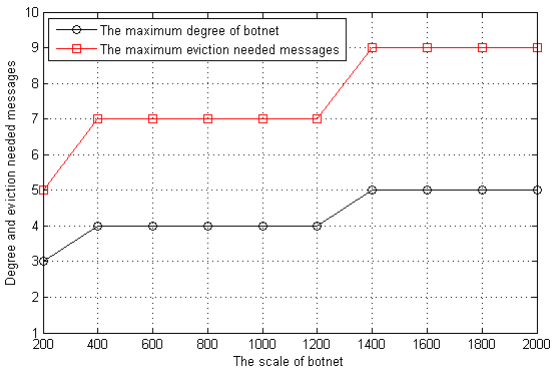


Figure 14 The relationship between the degree of botnet and eviction needed messages

We set the height of botnet is 5, the maximum degree of 200-botnet is 3, the maximum degree of 200-botnet is 5, the replacement needed messages of 200-botnet are 6, and the replacement needed messages of 2000-botnet are 10. Also, from Figure 15, we can see that the replacement of our botnode is easy and energy resource saved.

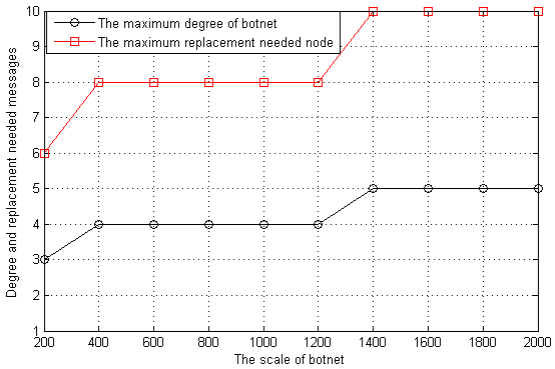


Figure 15 The relationship between the degree of botnet and replacement needed messages

The eviction and replacement period has strong operability, and the messages that needed of our botnet is acceptable, so we can say that our botnet has strong robustness.

VI. CONCLUSION

In this paper, we proposed the SMS based heterogeneous mobile botnet. The heterogeneous bot nodes and networks structure have made the C&C channel more efficient and secure. The communication protocol, botnet height and degree control mechanism, the node replacement of our mobile botnet is reasonable and acceptable. From the evaluation section, we can see that our mobile botnet has a highly connectivity, security and robustness.

Our future work includes the intensive study of C&C protocol of botnets, such as SMS-based C&C protocol and IP-based C&C protocol. We plan to study the authentication algorithm and encryption algorithm of bot node communications of our mobile botnet. Meanwhile, we would like to study the counter measures of SMS based mobile botnets.

ACKNOWLEDGMENT

The work described in this paper has been supported by Major Program of National Science and Technology (2011ZX03002-005-01), Key Project of Chinese Ministry of Education and National Key Technology R&D Program (2009BAH39B02), China Postdoctoral Science Foundation funded project (20100471762).

REFERENCES

[1] Geer D, "Malicious bots threaten network security," IEEE Computer, 2005,38(1):18-20.
[2] LiPeng Song, Jin Zhen, GuiQuan Sun, "Modeling and analyzing of botnet interactions," Physica A 390 (2011) 347–358.
[3] LiPeng Song, Jin Zhen, GuiQuan Sun. "Influence of removable devices on computer worms: Dynamic analysis

and control strategies," Computers and Mathematics with Applications 61 (2011) 1823–1829
[4] Lee WK, Wang C, Dagon D," Botnet Detection: Countering the Largest Security Threat," New York: Springer-Verlag, 2007.
[5] Norman ASA. Mobile phone threats - hype or (finally) truth? Security Articles -Archive, 2009. http://www.norman.com/security_center/security_center_archive/2009/67174/en, read: 2009.05.31.
[6] Irfan Asrar. Could sexy space be the birth of the sms botnet? Symantec, Internet blog, jul 2009. http://www.symantec.com/connect/blogs/ could-sexy-space-be-birth-sms-botnet, read: 2009.07.30.
[7] Ikee.B, http://www.symantec.com/security response /writeup.jsp?docid=2009-112217-4458-99.
[8] P.A. Porras, H. Saidi, V. Yegneswaran, "An Analysis of the iKee.B iPhone Botnet," in Proceedings of the 2nd International ICST Conference on Security and Privacy on Mobile Information and Communications Systems (Mobisec), May 2010
[9] A. Mehrotra and L.S. Golding. Mobility and security management in the gsm system and some proposed future improvements. Proceedings of the IEEE, 86(7):1480-1497, Jul 1998.
[10] BBOS_ZITMO.B http://about-threats.trendmicro.com/ Malware.aspx?language=us&name=BBOS_ZITMO.B
[11] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In Internet Measurement Conference, 2006.
[12] P.Traynor, M.Lin, M.Ongtang, V.Rao, T.Jaeger, P.McDaniel, and T.L.Porta, "On cellular botnets: Measuring the impact of malicious devices on a cellular network core," in Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'09)
[13] K. Singh, S. Sangal, N. Jain, P. Traynor, and W. Lee, "Evaluating bluetooth as a medium for botnet command and control," in Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2010).
[14] J. Serror, H. Zang, and J. C. Bolot. Impact of paging channel overloads or attacks on a cellular network. In Proceedings of the ACM Workshop on Wireless Security (WiSe), 2006.
[15] Anne Ruste Flø, Audun Jøsang. "Consequences of Botnets Spreading to Mobile Devices," short-paper Proceedings of the 14th Nodic Conference on Secure IT System (NordSec 2009).
[16] David Dagon, Guofei Gu, Cliff Zou, Julian Grizzard, Sanjeev Dwivedi, "A Taxonomy of Botnets," University of Central Florida, 2005.
[17] Anne Ruste Flø, Audun Jøsang. "Consequences of Botnets Spreading to Mobile Devices," short-paper Proceedings of the 14th Nodic Conference on Secure IT System (NordSec 2009).
[18] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol- and structure- independent botnet detection. In Security, 2008.
[19] Yuanyuan Zeng, Xin Hu, Kang G. Shin, "Design of SMS Commanded-and-Controlled and P2P-Structured Mobile Botnet", The University of Michigan, Ann Arbor, MI 48109-2121, U.S.A. 2009.
[20] P. Wang, S. Sparks, and C. C. Zou. An advanced hybrid peer-to-peer botnet. In USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07), 2007.
[21] R. Vogt and J. Aycock. Attack of the 50 foot botnet. Technical report, 2006.
[22] P.Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in IPTPS, 2002.

Guining Geng, male, was born in 1981. He received his Bachelor Degree in college of computer science and technology from QUFU Normal University in 2006, and received his Master Degree in college of computer science and technology from Harbin Engineering University in 2009. He is now a Ph.D Candidate in Beijing University of Posts and Telecommunications. His research interests are cryptography and Mobile Internet Security.

Guoai Xu, male, was born in1972. He received the Ph.D degree from School of Information Engineering, Beijing University of Posts and Telecommunications, Beijng, China, in 2001. He is now a professor in BUPT. His research interests include Information Security Management, Software security and so on.

Zhang Miao, male, was born in March, 1980. He received the Ph.D degree from School of Information Engineering, Beijing University of Posts and Telecommunications, Beijng, China, in 2007. He is now a teacher in BUPT. His research interests include Software security, Mobile internet and so on.

Yanhui Guo is a an associate professor in Beijing University of Posts and Telecommunications. She received her Ph. D. in Signal and Information Processing from Beijing University of Posts and Telecommunications in 2003. Dr.Guo was also a visiting scholar in the School of Information Sciences in the University of Pittsburgh in 2009. She has published a number of papers in international conferences and journals. She is interested in intelligent information processing and information content security.

Guang Yang, female, was born in 1980. She received her Bachelor Degree in college of computer science and technology from Heilongjiang University in 2003, received her Master Degree and Doctor Degree in college of computer science and technology from Harbin Engineering University in 2007 and 2009, respectively. Her research interests are Mobile Internet Security and WSN Security.

Cui Wei is an engineer in Information Center of Ministry of Science and Technology of the People's Republic of China. He received his Ph. D. in cryptography from Beijing University of Posts and Telecommunications in 2009. Cui wei has published a number of papers in international conferences and journals. He is interested in signature and information security.

Power-aware Small World Topology in Ad Hoc Networks

Yingjie Xia
Department of automation, School of Electronic, Information, and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou 310012, China
Email: xiayingjie@zju.edu.cn

Mingzhe Zhu
School of Computing Science, Simon Fraser University, Burnaby, B.C., V5A1S6 Canada
Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou 310012, China
Email: mingzhez@cs.sfu.ca

***Abstract*—Ad hoc network is a popular research topic in wireless network recently. It has good performance on the robustness and dynamics, such as minimal configuration and quick deployment for emergency situations. However, the ad hoc network also suffers from the limited power source provision because of its fundamental architecture. Therefore, the power issue becomes one of the most important concerns in ad hoc networks. In this paper, we propose to build a small world topology in the ad hoc networks as a topology control method to solve the power issues. Following the rule of small world topology, we manually construct a self-adjust power-aware small world network by implementing the decision method for random probability and action protocols to decrease the power consumption. Although this work is not the optimal solution to save power consumption in ad hoc networks, it can achieve a tradeoff between transmission efficiency and power consumption. And the self-adjustment feature of the small world topology can also facilitate the performance to prolong the lifetime of whole ad hoc network.**

***Index terms*—Ad Hoc Networks, Small World Topology, Random Probability, Head Node, Inner Node, Self-adjustment**

I. INTRODUCTION

The concept of commercial ad hoc networks arrived in the 1990s with notebook computers and other viable communications equipment, and it becomes a quite popular research topic recently. The history of ad hoc network can be traced back to 1972, and the DoD-sponsored packet radio network (PRNET) evolved into Survivable Adaptive Radio Networks (SURAN) program in the early 1980s. Ad hoc networks are suitable for use in situations where infrastructure is either not available, not trusted, or should not be relied on in times of emergency. A few examples include: battlefield; sensors scattered throughout a city for biological detection; an infrastructure less networks of notebook computers in a conference or campus setting; rare animal tracking; space exploration; undersea operations; and temporary offices such as campaign headquarters. It needs minimal configuration and can be quickly deployed for emergency situations.

The wireless ad hoc networks – also called WANET sometimes—are made up of a group of mobile units equipped with radio transceivers that wish to communicate with each other over wireless channels. The nodes in a WANET relay information for each other in a multi-hop fashion, as a centralized authority is absent. In this way, nodes in a WANET are self-organized to form a decentralized communication network that does not rely on any fixed infrastructures. The network is dynamic with nodes leaving and joining at any time. However, for a fixed instant of time, the wireless ad hoc network can be modeled as a graph where the number of neighbors that a node can establish wireless links to is known as the node degree. Being constructed mostly of battery operated devices, the capacity and lifetime of wireless ad hoc networks can be surprisingly low. For this reason, power efficiency is a very important consideration in designing ad hoc networks.

In this scope, we design an ad hoc network using small world topology, known also as “six degrees of separation” [1], to achieve power-efficient performance. We manually construct a self-adjust power-aware small world network by implementing the decision method for the random probability and action protocols to decrease the power consumption. This design can achieve a tradeoff between transmission efficiency and power consumption, thereby prolonging the lifetime of whole ad hoc network without much efficiency loss.

The rest of this paper is organized as follows: the Section II will show some related work on the power related technologies in ad hoc networks and the former researches of small world networks. Then in Section III we will implement our small world topology with power concerns in ad hoc networks. The simulation experiments and discussion will be described in Section IV. And the Section V will state the conclusion and the future work for this paper.

II. RELATED WORK

Being limited by the transmitter power of nodes and available energy resources, it is very important for ad hoc

doi:10.4304/jcp.7.1.244-251

networks to adopt energy efficient strategies to maximize network lifetime. As S. Mahfoudh and P. Minet mentioning in their survey [20], there are generally four primary ways to realize the goal of saving energy in ad hoc networks. In this paper, we mainly focus on the topology control methods. Many topology control algorithms have been proposed to reduce energy consumption and improve network capacity, while maintaining network connectivity. The key idea of topology control is that instead of transmitting using the maximal power, each node adjusts its power transmission. Thus, energy dissipated in transmission is reduced and a new network topology is created. RNG, Related Neighborhood Graphs [2], removes any edge directly connecting two nodes if there is a path of two hops or more between them. MST, Minimum Spanning Tree [3], transforms the network graph in a minimum energy broadcast tree rooted at a given source node giving the shortest path to any other node; only the energy dissipated in transmission is taken into account .LMST, Local Minimum Spanning Tree [4], is a localized minimum spanning tree where each node computes its own MST in its neighborhood and only retains those one hop neighbors in the tree as its neighbor in the final topology. In [5], the idea is to reduce the transmission range of every node to minimize energy dissipated in transmission. However, these algorithms all focus on pure power saving research without considering other important aspects in wireless transmission of ad hoc networks.

To have a wider concern on system performance, other important features should be involved, and even to achieve tradeoff among them. For example, in [6], the TCH (Topology Control with Hitch-hiking) is used to obtain a strongly connected topology minimizing energy dissipated in transmission with partial signals. It not only does consider the power issues, but also involves the topology connection as a important part in research. In [7], Adaptive Transmission Power Control (ATPC), is proposed which allows each node to know the optimal transmission power level to use for each neighbor while maintaining a good link quality. In this paper, we concern the additional feature, the performance of efficiency, besides the power consumption. We will use the small world network to implement the topology control to facilitate the performance on power saving and transmission efficiency in ad hoc networks.

The small world phenomenon became famous since the experiment held by Stanley Milgram [8], in which the six degrees of separation is observed. Generally, it is a topology network in which the nodes can connect to any other ones within average limited hops (or paths). In early days, there is much work on the theoretical research of small world. Duncan Watts and Steve Strogatz [9] showed the theoretical description and analysis on the small world network, which is neither completely regular nor completely random, but a case between these two extremes. They also donated some notations such as random probability, path length and cluster coefficient which are very helpful for the research on the small world network. Kleinberg [10, 11] proposes a theoretical framework for analyzing graphs with small world properties. His work reveals that the small world model has two fundamental components: first short chains exist ubiquitously, and second individuals operating with purely local information are very adept at finding these chains.

Recently, there is much research on the practice of small world network. Merugu et al. [12] propose a small world overlay structure on top of an unstructured and decentralized P2P network, while Hui, Lui and Yau [13] implement such a model over the structured and decentralized P2P network. The common feature of both work is to provide a small world topological structure, so as to efficiently look up and transfer resources in P2P systems. Xia et al. [18] designed SW-R2P, a trusted small world overlay P2P network with role based and reputation based access control policies, in order to implement an efficiency and security Peer-to-Peer network. All the work above focuses on the small world topology used to solve P2P issues, such as robustness, hop length and reliability. Similarly, there is also some research concerning on small world topology in ad hoc networks. Sonja Filiposka et al. [14] conduct a research on the small world in ad hoc networks which has an extreme influence on enhancing the social characteristics related performance. There is also a small world in motion (SWIM) implemented by Sokol Kosta, Alessandro Mei and Julinda Stefa [15] to improve the performance of the ad hoc mobile networks. Helmy[16] and Dixit et al [17] proposed the construction of wireless ad hoc networks with small average path length by adopting the small world paradigm. However, all the research work above focuses on the feasibility, efficiency and robustness, ignoring the power issues which should be such an important concern in ad hoc networks. In this paper, our model considers the power issue with self-adjustment strategy to achieve tradeoff between efficiency and power consumption.

III. SMALL WORLD AD HOC NETWORK IMPLEMENTATION

The small world topology evolved in ad hoc networks can be shaped as the topology control method going on between clients. It is usually not an inherent paradigm. In order to improve the performance of transmission efficiency, network robustness and power persistency in ad hoc networks, the network topology can follow the principle to be constructed and adjusted as a small world paradigm. Therefore, how to construct and maintain a small world topology in ad hoc networks is the key problem to be solved.

The implementation for the small world ad hoc networks may be different on variable cases. We need to create some principles with which to determine the probability p and the protocols how we can construct the short links and long links. The details will be elaborated to show the implementation of small world ad hoc networks.

A. Preliminary Terminologies

Firstly, we clarify some important notations of the small world network, which can be categorized into link roles and node roles.

The link roles include long link and short link (or inner link). The long link is used to connect different clusters of nodes, and the short link is responsible for the connection in the cluster. The principle for the short link construction is simple, while the principle for the long link construction which is determined by the link probability p is more complex.

The nodes roles include three types of nodes: head node, candidate head node and inner node. The head node takes main responsibility for connecting clusters based on some probability. It is very important in expanding the scale of whole network. The candidate head node is the shadow node of head node, which means it may take the place of head node when the original head node leaves, fails or becomes unaccessible. It is required to backup some data of the head nodes used to recover the network when the head nodes need to be substituted. The inner nodes are the main participants in the cluster. They are connected by short links within the cluster. Based on the requirements of network, we can evaluate the performance of the nodes to determine the head nodes, candidate head nodes and inner nodes.

B. Principle to determine probability p

The first step in small world network implementation is to determine the probability p for long link creation. We set the requirements or the concerns of network, and formulate functions for them as parts to construct the probability p. They have different scales which need to be mapped to a consistent range. Finally we add them after multiplying corresponding weights. The formula is defined as follows.

$$p = F(w_1 f_1(x_1) + w_2 f_2(x_2) + w_3 f_3(x_3)) \quad (1)$$

After determining the principles and functions to calculate the probabilities, as the second step, head nodes can evaluate their long link probabilities with nodes in other clusters. Choosing proper nodes to build a long link expands the scale of small world network. The head node will calculate the probability p based on its prefetched knowledge, and decide the long link state with that probability. As a simple implementation, we will randomly generate a number, if it is in the range of the probability, then we build the link, otherwise we will not. We may also update the probability if it is necessary, while the network is running.

We use a simple decision model for the probability p in our power-aware small world. It is consist of multiple facets, including the power condition (PC), the number of Links held already (LN), the distance to the target nodes (ND), and an overall decision (OD) for the probability p. Figure 1 shows the model structure.

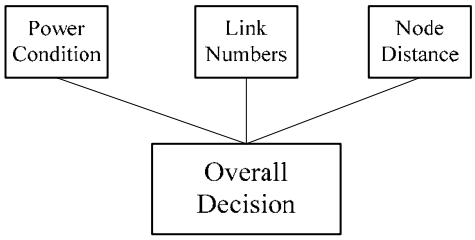


Figure 1. Decision model specialized for power-aware small world ad hoc networks

- PC is the metric denoting the power condition of nodes, mostly head nodes. It can be calculated automatically in the intercommunication between nodes. We formulate it as $PC = Power_{current} / Power_{full}$ where $Power_{current}$ and $Power_{full}$ are the power values for current condition and full-charged condition. In this way, we normalize the PC of devices in ad hoc networks to the range from 0 to 1.
- LN is the metric denoting the long link numbers which nodes hold in the interaction. This metric is included for taking a consideration in power saving. If the node holds enough long links, we need not to build other long links to expand the range as well as the power. The normalizing for LN is not as simple as PC, in which we formulate a function mapping link numbers to the range from 0 to 1. The function should be defined as mapping link number 0 to value 1, and through the link numbers increasing, the values are more close to 0. Finally, we choose the exponentially decreasing function $LN = k^{-x}$, in which k is a real number greater than 1. We can choose a proper k for a corresponding application depending on its scale of ad hoc network and expectation of long link number.
- ND is the metric denoting the distance of two nodes in the interaction. It is significantly related to the power consumption through the inverse square law, $E \propto I / d^2$. The normalizing function should maintain the relationship between the power and the distance, and is better to be practical. We finally choose $ND = (d_0 / d)^2$, where d_0 is the distance to the closest node which is not in the cluster. This is reasonable because the closest one out of the cluster is the most power efficient node to link, which should be mapped to the largest value 1 for the distance.
- OD is the final result of the decision network, which comprehensively evaluates all the leaf metrics introduced above. We can calculate it by the function $OD = W_{PC} * PC + W_{LN} * LN + W_{ND} * ND$, where W_{PC}, W_{LN}, and W_{ND} are the weights to the metrics respectively, and $W_{PC} + W_{LN} + W_{ND} = 1$. The weights can simply be determined by the head node of the cluster according to its practical concerns. OD is regarded as the probability which head node uses to create long links to other cluster. The head nodes

hold OD instances for corresponding clusters, which will be applied to some key operations, such as maintaining the node list for long links, searching new nodes for routing, etc.

This model design for the ad hoc networks depends on their properties that the nodes can be both routers and terminators. Therefore, we can design a principle for each single node, and drive the network topology evolving into a small world network.

The probability implementation is treated as the evaluation in multiple differentiated power aspects of nodes. It can be customized by adding new aspects or constructing a more complicated decision network topology. During the network construction or data transmission, the resource providers can be selected based on their trust evaluations. The requester can combine several aspects of the candidate providers into the overall probability which decides to create appropriate long links. For example, the boundary case is that when the header node's PC is full and its LN is zero, if it can find the closest node to it, there is no reason to reject that link. And in this case, the probability for that decision model is just 1.

C. Criteria for head node

When a new node joins one of the clusters in the network, it is necessary to determine its role in the cluster, head node, candidate head node or inner node. We will show criteria for the cluster to choose the head node and candidate head node. The criteria is used to judge whether the node can afford to play the role as the head nodes based on its power related and other properties. The criterion function is as following:

$$C(node) = w_1 Power(node) + w_2 Seen(node) + w_3 Range(node) \quad (2)$$

In this function, *C()* is the overall evaluation for a node, and w_1, w_2, w_3 are weights for different concerns. The *Power()* represents the node's power capability, the *Seen()* is the number of other nodes covered with by the node under its maximum radio, and the *Range()* is the maximum range that the node can reach.

As following the function above, the original ad hoc network may gradually evolve to a small world ad hoc network. Furthermore, to achieve the power-aware small world implementation, we still need to design specific protocols about the action on the nodes and links.

D. Protocols Implementation

In this section, we will describe protocols for building and maintaining the power-aware small world topology in ad hoc networks. The action protocols are specifically for the events of node joining a cluster, leaving a cluster and failing.

(1) Node Join Protocol (NJP)

If node i requires to join a cluster g, we need to determine its role in the network. For a rookie node, which means it has no interaction with any other node or it transfers from one cluster to another cluster, the node can only use its criteria value in the cluster g to evaluate its capability for node role. By comparing the criteria value of node i with j which takes a special node role in a cluster, such as the head node or the candidate head node, we can determine the node role of i. If the value of node i is greater than j, i will join the cluster g and replace the role of j to be a head node or a candidate head node. Otherwise, j will keep its role and i will be set as the inner node.

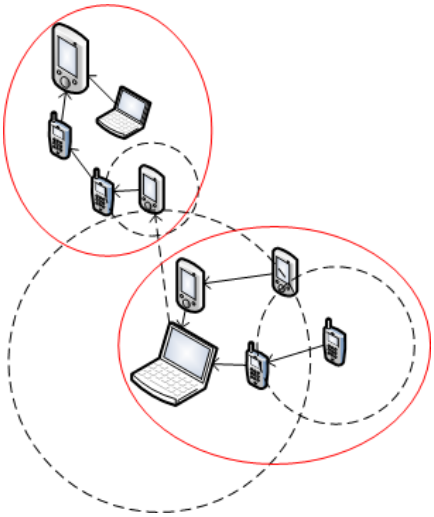


Figure 2. Short link construction for a new joining

However, before the node i sends the criteria, it firstly needs to construct the short links to reach other nodes. In the short link construction, we achieve a method which can avoid the power law phenomenon and have a benefit to save power based on the small world. The short links in this protocol are not simply connected to the head nodes. If a node i joins the network, it will increase the range to find a head node, or node which can reach the head node in limited hops (2 or 3 hops) as Figure 2 shows. Then it will join the cluster g of that head node, and send its criteria value to determine its role.

The algorithm of the node join protocol is specified in Algorithm 1 below.

Algorithm 1: Node Join Protocol

INPUT i : joining node; g : joined cluster;
j : special role node of g
OUTPUT g' : the new cluster i joins
GL_i =Initialize(i) //initialize i 's short links
R_i = inner node of g // initialize i 's role as inner node
$g' = g + i$ // add node i to cluster g
$C(i) = w_1 Power(i) + w_2 Seen(i) + w_3 Range(i)$ // calculate i 's value
If $C(i) > C(firstCandidateheadnode)$:
For all head node j
$C(j) = w_1 Power(j) + w_2 Seen(j) + w_3 Range(j)$ // calculate j 's value
If ($C(i) > C(j)$)

R_i = head node of g //set i 's role as head node
GL_i += Update(GL_j) //add possible short links to i
LL_i = Update(LL_j) //set i 's long link
GL_j = initialize(j)
break
Else
GL_j += short link to i //add a short link from j to i
EndIf
Endfor
Else
For all candidate head node j
$C(j) = w_1 Power(j) + w_2 Seen(j) + w_3 Range(j)$
// calculate j's value
If ($C(i) > C(j)$)
R_i =candidate head node of g // set i as candidate
$data(LL_i)$ = Replicate($data(LL_j)$) // copy j's long link data
Message(Head node, R_i) // update i 'role in cluster
break
EndIf
Endfor
Endif
Output g'

(2) Node Leave Protocol (NLP)

When a node i leaves the cluster g, it performs differently according to its role. If it is a head node, the first mission it should complete is to find a substitutor node j in its cluster g, which holds the evaluation closest to i. In our algorithm, to find such a node is similar to choose the candidate head node. Head node i sends a message to all the other nodes in its cluster, notifying that leaving the cluster and choosing the candidate head node j as the new head node. Then all the nodes in the cluster will calculate their hops to the new head node to adjust their short links. The new head node usually holds a replicated data from the former head node. It can update those data and replicate them to a new candidate head node. The nodes which originally have short links to that leaving head node need to modify its range and rearrange their short links.

There are two situations for i to leave the cluster g, which are i just leaving the network and i joining another cluster in the network. If the cluster g contains only one node i and i leaves the network, the cluster is eliminated from the network straightforwardly. If the cluster g contains more than one nodes and the substitutor j is picked out, it needs to complete the whole process for the head node leaving. In the other situation, if i leaves the cluster g and joins another cluster, the implementation of the leaving part is almost the same as the first situation, and the implementation of the joining part can be handled by NJP.

We implement the node leave protocol in such a way that it can keep the whole system stable and consistent. Its algorithm is introduced in detail as following.

Algorithm 2: Node Leave Protocol

INPUT i : leaving node; g : i 's cluster
j : candidate head node of g
OUTPUT g' : the cluster i leaves; g'' : the cluster i joins
If ($sizeof(g) == 1$) // cluster g contains only one node
g' = NULL // cluster g is set empty
If ($Action(i)$ == "join another cluster") // i joins another cluster
NJP (i , g'') // apply node join protocol to i joining g''
Endif
Else // cluster g contains more than one node
R_j = head node of g
GL_j += Update(GL_i) // add possible short links to j
LL_j = Update($data(LL_j)$) // set j's long link from its data
Broadcast(i , j) // update other nodes
If ($Action(i)$ =="join another cluster")
// i joins another cluster
NJP(i , g'') // apply node join protocol to i joining g''
g' = $g - i$ // Remove i from cluster g
Endif
Output g' , g''

(3) Node Failure Protocol (NFP)

Each node periodically sends request to the head node of its cluster to state it is alive and to judge whether the head node is alive. If a node i does not respond to other node's request beyond the time threshold, i is considered as a failed node, and the network loses all the information of i. In that case, the network uses the node failure protocol to recover and maintain the remained links.

Supposing the node i is in the cluster g, if g contains only one node i who fails, g becomes empty and is eliminated from the network automatically. The elimination will be detected by other nodes which link to g and remove their corresponding long links later. If there are more than one node in the cluster g and the failed node i is the head node of the cluster, we find a substitutor node j which is the candidate head node in that cluster g. The way to find the node j is the same as to find a substitutor for a leaving node which is specified in NLP. In another situation, if node i is an inner node, then it does not need a substitutor but has to call those nodes which has short link to i to update the short links similar to NLP.

The node failure protocol stabilizes the network when a node failed, and keeps the network robust to various attacks. Algorithm 3 describes NFP.

Algorithm 3: Node Failure Protocol

INPUT i : failed node; g : i 's cluster

OUTPUT g' : the cluster after i is failed and removed

If ($sizeof(g) == 1$) // g contains only one node

g' = NULL // cluster g is set empty}

Else // g contains more than one node

If (R_i == "head node") //node i is an head node

R_j = head node of g

GL_j += Update(GL_i) //add possible short links to j

LL_j = Update($data(LL_j)$) //set j's long link from its data

Broadcast(i , j) // update other nodes

Else

Broadcast(i)// update other nodes sent by head node

Endif

$g' = g - i$ //Remove node i from cluster g

Endif

Output g'

IV. Experiments and Discussion

A. Simulation Setup

Our simulation experiments are based on rewriting the Matlab simulator Prowler which implements a global topology control and a dynamic node-driven self-adjust topology control.

The Matlab simulator Prowler is deployed on the lab PC as the simulation environment. The PC is set up with Intel Dual Core 2.13GHz and 2G memory. We simulate 100 nodes with different initialized power condition range from 1 to 10. The nodes will be randomly deployed in area, and we assume a consistent distance measure for calculating the power consumption.

B. Experiments

We use three simulation experiments to evaluate the performance of our ad hoc networks in the power-aware small world topology. We compare our model with other topology and routing models, such as spantree and clusterhead gateway switch routing (CGSR) [19]. The spantree model is used to show the advantages of our model on topology-related efficiency. And the power-unaware small world model, CGSR, is chosen to explore the advantage of our model on energy saving for our power-aware small world topology can be regarded as an extension of the CGSR on power concerns and self-adjusting functionality.

(1) Power Consumption

The first experiment is about the power consumption. We design it to compare our model with spantree model and CGSR model. Assuming that a random pair of nodes request for communication continually and a node which relay the communication consumes 1 power, the experiment is designed as that along with the time how much the power of the whole network is consumed. The results are shown in Figure 3, with the time as the x-coordinate and the total power of the network as the y-coordinate.

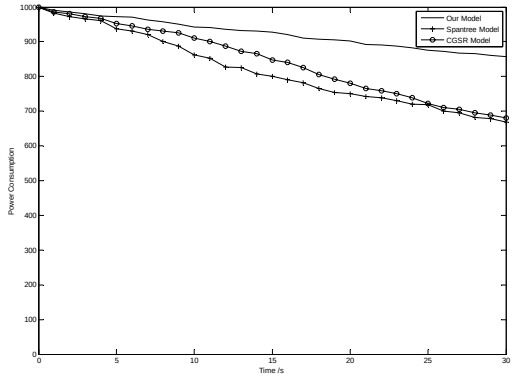


Figure 3. Power consumption experiment on three models

The experimental results show that all three curves decrease, and our small world model decreases more slowly than the other two models. This is caused by that in general our model is a greedy optimal method. We try to minimize the power consumption on every node through choosing proper headnode and links. Even that it has not been proved theoretically our method can achieve global optimal, the experiment still shows its better results than other two models who have no efficient energy control approach.

(2) Average Path Length

The second experiment is about the average path length, which is an important evaluation to compare our model with spantree model and CGSR model. We increase the number of nodes in the network from 50 to 150. Each node randomly chooses another node to communicate with, and we calculate the average path length of the whole network. The results are shown in Figure 4, with the number of nodes as x-coordinate and the average path length as y-coordinate.

From the experimental results, we observe that from 50 nodes to 150 nodes the average path lengths for our model and CGSR model are all less than 5, and their increments also do not exceed 1. However, the compared

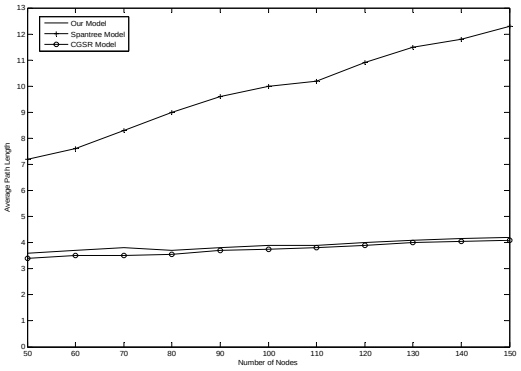


Figure 4. Average path length experiment on three models

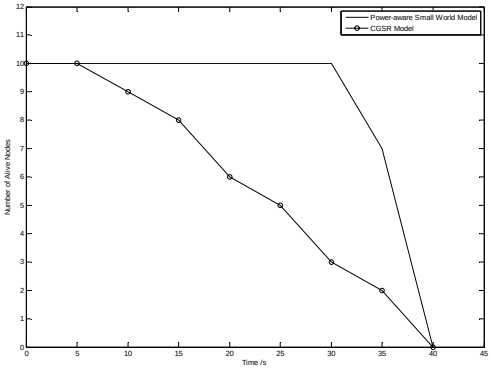


Figure 5. Life time experiment on two models

spantree model has greater lengths and increases much faster. These indicated advantages are mainly brought about by their small world topology. Although the average path length of our model is quite close to CGSR model, it is still a little bit greater than CGSR. This is caused by multihops intra clusters for power concerns.

(3) Life Time

The third experiment is about the life time of the whole ad hoc network. Through this experiment, we want to show that by taking the power condition into the decision model we can keep a power consumption balance among all the nodes, therefore we can keep the life time of the network as long as possible. Our model is compared with the CGSR model, power-unaware small world model. The experiment is designed as that along with the time how many nodes alive in the network. The results are shown in Figure 5, with the time as the x-coordinate and the number of alive nodes in a cluster as the y-coordinate.

The experimental results show that the two models have the same life time, however during the life time the curve of power-aware small world model is more balanced than the curve of power-unaware small world model. This is because the power-aware model can do the self-adjustment based on the power condition of the nodes to make them alive as long as possible. The CGSR model can easily fall into the structure with power law phenomenon, which is good for decreasing hop numbers but is lack for saving and balancing energy consumption in networks, especially for ad hoc network with more stable structure applications. For example, when one head node's power is under the threshold which can be set by the system, our model will choose another node with more power to take the position of head node, and reduce the scale and the links to save the power of the original head node. However, in CGSR model draining out one single node may cause isolated nodes in the ad hoc network and some other related problems.

V. Conclusions and Future Work

In this paper, we use the power-aware small world topology as an adaptive power efficient strategy to maximize network life time and efficiency of the ad hoc networks. We design a kernel decision model for the principles of the random probability and head node determination. We also implement the action protocols, such as the NJP, NLP and NFP with a self-adjustment scheme for the power consumption balance.

Finally, we use the simulation experiments to test the performances on power consumption, average path length and life time. The results show that our model can achieve a tradeoff between transmission efficiency and power consumption, and also hold the self-adjustment feature to facilitate the performance to prolong the lifetime of whole ad hoc network.

For the future work, we may optimize our implementation according to other theories. For example, we can use some more sophisticated approaches such as the Bayesian decision theory to determine the random probability p of the long links. We can also add more action protocols to specify diverse behaviors of the nodes. Finally, we will try to move our implementation from the simulation experiments to the practical applications which can be used to test the feasibility and practicability of our work.

Acknowledgement

The authors thank Project 61002009 supported by National Natural Science Foundation of China, Project 2010C31018 supported by Science and Technology Planning Project of Zhejiang Province, and Scientific Research Fund of Hangzhou Normal University under grant number HSKQ0042.

References

[1] J. Guare, *"Six Degrees of Separation: A Play"* Vintage Books, New York, 1990.

[2] G. Toussaint, "The relative neighborhood graph of a finite planar set", *Pattern Recognition*, Vol 12, 1980, doi:10.1016/0031-3203(80)90066-7.

[3] J. Wieselthier, G. Nguyen, A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks", *INFOCOM 2000*, Tel Aviv, Israel, March 2000, doi:10.1109/INFCOM.2000.832232.

[4] N. Li, J. Hou, L. Sha," Design and analysis of an MST-based topology control algorithm", *INFOCOM 2003*, San Francisco, CA, April 2003, doi:10.1109/TWC.2005.846971.

[5] F. Ingelrest, D. Simplot-Ryl, I. Stojmenovic, "Optimal Transmission Radius for Energy Efficient Broadcasting Protocols in Ad Hoc Networks", *IEEE Transactions on Parallel and Distributed Systems*, June 2006, doi: 10.1109/TPDS.2006.74.

[6] M. Cardei, J. Wu, S. Yang, "Topology Control in Ad hoc Wireless Networks with Hitch-hiking", *First IEEE SECON04*, October 2004, doi:10.1109/SAHCN.2004.1381950.

[7] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks", *SenSys 2006*, Colorado, November 2006, doi:10.1145/1182807.1182830.

[8] S. Migram, "The small world problem", *Psychology Today*, vol 2, pp60–67, 1967.

[9] D. Watts, S. Strogatz, "Collective dynamics of 'small-world' networks". *Nature* 393 (6684): pp. 440-442, 1998, doi:10.1038/30918.

[10] J. M. Kleinberg, "Navigation in the small world", *Nature* 406, p. 845, 2000, doi:10.1038/35022643.

[11] J. Kleinberg, "*The Small World Phenomenon: an Algorithmic Perspective*", Technical Report, 2000, doi: 10.1145/335305.335325.
[12] S. Shashidhar Merugu and E. Zegura, "Adding Structure to Unstructure Peer-to-Peer Networks: the Use of Small World Graphs", *Journal of Parallel and Distributed Computing*, 65(2):142-153, 2005, doi:10.1016/j.jpdc.2004.09.004.
[13] K. Hui, J. Lui, and D. Yau, "Small-World Overlay P2P Networks: Constructing, Management and Handling of Dynamic Flash Crowd," *Computer Networks*, vol. 50, pp. 2727-2746, 2006, doi:10.1016/j.comnet.2005.10.010.
[14] S. Filiposka, D. Trajanov, A. Grnarov, "Analysis of small world phenomena and group mobility in ad hoc networks", *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*, Book chapter, Springer Netherlands, 2007, doi:10.1007/978-1-4020-6266-7_77.
[15] S. Kosta, A. Mei, and J. Stefa, "Small World in Motion (SWIM): Modeling Communities in Ad-Hoc Mobile Networking", *IEEE SECON 2010*, Boston, MA, USA, 2010, doi:10.1109/SECON.2010.5508278.
[16] A. Helmy, "Small worlds in wireless networks," *IEEE Communications Letters*, 7(10):490-492, 2003. doi:10.1109/LCOMM.2003.818887.
[17] S. Dixit, E. Yanmaz, and O. Tonguz, "On the design of self-organized cellular wireless networks," *IEEE Communications Magazine*, July 2005, doi: 10.1109/MCOM.2005.1470827.
[18] Y. Xia, G. Song, and Y. Zheng, "SW-R2P: A Trusted Small World Overlay P2P Network with Zero Knowledge Identification", Journal of Computers, 3(10):3-11, 2008, doi:10.4304/jcp.3.10.3-11.
[19] E. Royer, and C. Toh, "A Review of Current Routing Protocols or Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications*, April 1999, doi:10.1109/98.760423.
[20] S. Mahfoudh, and P. Minet, "Survey of Energy Efficient Strategies in Wireless Ad Hoc and Sensor Networks," *IEEE Seventh International Conference on Networking*, pp. 1-7, Cancun, Mexico, 2008, doi:10.1109/ICN.2008.55.

Yingjie Xia was born in Fenghua, Zhejiang Province, P.R. China on November 7th, 1982. Now he is a Postdoc in Department of automation, School of Electronic, Information, and Electrical Engineering, Shanghai Jiao Tong University, and an instructor in Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou, Zhejiang, P.R. China. He was sponsored by China Scholarship Council as a visiting scholar in University of Illinois at Urbana-Champaign (UIUC), USA. He received his Ph.D. Degree in the College of Computer Science, Zhejiang University in 2009. His major is computer science and technology.

His research interests cover distributed computing, grid computing and P2P network. He deployed the Zhejiang University Campus Grid in 2006 by hierarchically combining the Globus Toolkit and Sun Grid Engine. And he also implemented a R2P system, P2P system with role and reputation based access control. His work focused on to improve the R2P system and proposed a SW-R2P model, a trusted small world P2P network in Zhejiang University. Now he tries to do some research in ad hoc network.

Mr. Xia has published more than 30 papers.

Mingzhe Zhu is a student in School of Computing Science, Simon Fraser University, and a Research Assistant in Center for Service Engineering, Hangzhou Normal University. He received the Bachelor degree from the Zhejiang University in 2008. He specializes in P2P, distributed computing and cloud computing.

Learning Rates of Support Vector Machine Classifiers with Data Dependent Hypothesis Spaces

Bao-Huai Sheng
Department of Mathematics, Shaoxing College of Arts and Sciences Shaoxing, Zhejiang 312000, China
e-mail: shengbaohuai@ 163.com

Pei-Xin Ye
School of Mathematics and LPMC, Nankai University,Tianjin 300071, China
e-mail: yepx@nankai.edu.cn

***Abstract*—We study the error performances of p -norm Support Vector Machine classifiers based on reproducing kernel Hilbert spaces. We focus on two category problem and choose the data-dependent polynomial kernels as the Mercer kernel to improve the approximation error. We also provide the standard estimation of the sample error, and derive the explicit learning rate.**

***Index Terms*—Support vector machine classification; Learning rate; Reproducing kernel Hilbert spaces; Cesaro means.**

I. Introduction and Results

Support vector machine classification [1]-[7], [9]-[25] has a foundation in the framework of statistical learning theory and classical regularization theory for function approximation. It is one of the most important topics in the field of machine learning. It has been applied successfully to various practical problems in science, engineering and many other related fields. The goal of classification is to construct a classifier which can predict the unknown class of an observation with small misclassification error. This problem has been studied widely and many important algorithms have been developed (Ref. [3]-[8]).

Let $X=[-1,1]$, $Y=\{-1,1\}$. A binary classifier $f: X \to Y$ divides the input space X into two classes.

Let ρ be an unknown probability distribution on $X \times Y$ and $(\mathcal{X}, \mathcal{Y})$ be the corresponding random variable. The misclass -ification error for a classifier $f: X \to Y$ is defined to be the probability of the event $\{f(\mathcal{X}) \neq \mathcal{Y}\}$

$$\mathcal{R}(f) := \text{Prob}\{f(\mathcal{X}) \neq \mathcal{Y}\} = \int_X P(\mathcal{Y} \neq f(x) \mid x) d\rho_X(x),$$

where ρ_X is the marginal distribution on X and $\rho(y \mid x)$ is the conditional probability measure at x induced by ρ . The distribution ρ is known only through a set of samples $z := \{z_i\}_{i=1}^m = \{(x_i, y_i)\}_{i=1}^m \in Z^m$ independently drawn according to ρ .It is known from [9] the classifier which minimizes the misclassi -fication error is the Bayes ruler $f_c := \text{sgn}(f_\rho)$, where

$$f_\rho(x) = \int_Y y d\rho(y \mid x) = P(y=1 \mid x) - P(y=-1 \mid x), \quad x \in X.$$

Denote the p -norm hinge loss function as

$$V(yf(x)) := (1-yf(x))_+^p = |y - f(x)|^p \, \chi_{\{yf(x) \le 1\}}, \quad (1)$$

where $\chi(\cdot)$ is the indicator function. $V(yf(x))$ measures the cost paid by replacing the true y with the estimate $f(x)$. The corresponding V -risk is

$$\mathcal{E}(f) := \int_Z V(yf(x)) d\rho = EV(yf(x)).$$

Let $f_\rho^V : X \to R$ be a measurable function minimizing the expected risk $f_\rho^V := \arg\min \mathcal{E}(f)$, where the minimum is taken over all measurable functions. According to [8], we may always choose a f_ρ^V such that $f_\rho^V(x) \in [-1,1]$ for each $x \in X$. Since the expected risk involving the unknown distribution ρ is not computable, its discretization is used instead which is computable in terms of the sample z, is defined as

$$\mathcal{E}_z(f) := \frac{1}{m}\sum_{i=1}^m V(y_i f(x_i)) = \frac{1}{m}\sum_{i=1}^m (1-y_i f(x_i))_+^p. \quad (2)$$

Regularized learning schemes are implemented by minimizing a penalized version of the empirical error over a set of functions, called a hypothesis space. Then the regularized classifier generated for a sample $z \in Z^m$ is defined as $\text{sgn}(f_{z,\lambda})$, where $f_{z,\lambda}$ is a minimizer of the following well-known Tikhonov regularization scheme

$$f_{z,\lambda} := \arg\min_{f \in \mathcal{H}} \Big\{ \frac{1}{m}\sum_{i=1}^m V(y_i f(x_i)) + \lambda \Omega(f) \Big\}. \quad (3)$$

Here λ is called regularization parameter, it depends on m and usually $\lambda(m) \to 0$ as m becomes large. This

Corresponding author: Ye Peixin. This work is supported by the Natural Science Foundation of China (Grant No. 10871226, 10971251.)

doi:10.4304/jcp.7.1.252-257

kind of scheme is very popular in many areas of the applications and theory of machine learning.

In this paper, we take the hypothesis space H to be the reproducing kernel spaces reproducing by polynomials on $[-1,1]\times[-1,1]$ and a given data in $[-1,1]$. We want to estimate excess misclassification error $\mathcal{R}(\mathrm{sgn}(f_{z,\lambda}))-\mathcal{R}(f_c)$. Many investigations of this topics have been done when the kernel $K(x,y)$ is the Gaussian kernel $K_\sigma(x,t)=\exp\{-\frac{\|x-t\|^2}{2\sigma^2}\}$ (Ref. e.g.[5],[8]). Now let us turn to other kernels, when the kernel is a polynomial kernel, the excess misclassification error estimate may become simpler since the polynomial class has many advantages in estimating the covering number, which is needed in presenting the error estimate. In fact, the excess misclassification error estimate for the polynomial kernel is a field investigated by many mathematicians. Among others, [5] gave a quantitative estimate for the convergence rate in the univariate case $X=[0,1]$ with the Bernstein- Durrmeyer polynomials operators, [4] gave the corresponding estimate in the case $X\subset R^n$ being a simplex. It is known that the best error of the Bernstein operator is $O(\frac{1}{\sqrt{m}})$. Thus we introduce the generalized Vallee Poussin means of orthonormal algebraic polynomials which can approximate the target function with $O(\frac{1}{m^\alpha})$ for some $\alpha>0$, and the orthogonal algebraic polynomials may exist for any positive measure ρ_X (Ref.e.g.[12],[13]). These properties is much better than the the Bernstein operator and it enable us to estimate the excess misclassi -fication error by generalized Vallee Poussin means. So we may yield better approximation error and therefore the learning error can be improved.

Note that now our hypothesis space depends on the input data. This makes the analysis of the error quite difficult and different from the previous results which for the algorithms with a data independent hypothesis space (see [1]-[9]). There are already some literatures in this area but the research is not very rich yet. In [14] the uniform convergence inequality is studied for the data dependent functions. In the coefficient regularization was analyzed under the restriction that the kernel is positive semi-definite or has certain smoothness condition (such as Lipschitz condition).

Throughout the paper, we shall write $A=O(B)$ if there exists a constant $C>0$ such that $A\le CB$. We write $A\sim B$ if $A=O(B)$ and $B=O(A)$. Let $X=[-1,1]$ and assume the marginal distribution $d\rho_X(x)=w(x)dx$ with $w(x)=(1-x)^{\alpha_1}(1+x)^{\alpha_2}$, $\alpha_1>-1,\alpha_2>-1,$ $\alpha_1+\alpha_2>-1,$ being Jacobi weights on $[-1,1]$ with finite moments; i.e., $\int_{-1}^1|x|^k\,w(x)dx<\infty,\quad k\in\mathrm{N}_0:=0,1,2,\cdots$. It is known from [14] that for all $k\in\mathbb{N}_0$ there exists a unique polynomial

$$p_k(w,x)=\gamma_k(w)x^k+\cdots,\qquad \gamma_k(w)>0$$

such that

$$\int_{-1}^1 p_k(x)p_{k'}(x)w(x)dx=\delta_{k,k'}.$$

Then we denote

$$a_k(f)=\int_{-1}^1 f(x)p_k(x)w(x)dx,\ k\in\mathrm{N}_0. \qquad (4)$$

Moreover, there uniquely exists Lagrange polynomial interpolating operator $L_n(x)$ of order $n-1$, such that

$$L_n(x_{k,n})=y_k,\qquad k=1,2,\cdots,n, \qquad (5)$$

for any real numbers y_k, $k=1,2,\cdots,n.$.

Let $p\ge1$, $L_p(dw)$ be the class of all measurable real functions f for which $\|f\|_{L_p(dw)}=\left(\int_{-1}^1|f(x)|^p\,w(x)dx\right)^{\frac{1}{p}}$ $<+\infty$. For $\delta>\max(\alpha_1+\frac{1}{2},\alpha_2+\frac{1}{2})$ the Cesaro means $C(\delta)$ of f is defined by

$$\sigma_N^\delta(f,x)=\frac{1}{A_m^\delta}\sum_{k=0}^N A_{N-k}^\delta a_k(f)p_k(x),\ x\in[-1,1],$$

$$A_k^\delta=\frac{\Gamma(k+\delta+1)}{\Gamma(k+1)\Gamma(\delta+1)},N=0,1,\cdots$$

For a given $\beta>0$ we define,

$$D(\beta)f\sim\sum_{k=1}^\infty k^\beta a_k(f) \qquad (6)$$

and we say $D(\beta)f\in L_p(dw)$ if there exists $\varphi\in L_p(dw)$ such that $a_k(\varphi)=k^\beta a_k(f)$. It is known form [17] that there is some $\delta>1/2$ such that

$$\|\sigma_N^\delta(f)\|_{L_p(dw)}\le C_2\|f\|_{L_p(dw)},\quad 1\le p\le\infty, \qquad (7)$$

with C_2 independent of N.

For $f\in L_p(dw)$ we define the generalized vallee Poussin means of the ortho-normal $\{p_k(x)\}$ by

$$\eta_N(f)(x)=\sum_{k=0}^{2N}\eta(\frac{k}{N})a_k(f)p_k(x),\ \forall x\in[-1,1], \qquad (8)$$

where $\eta(u)\in C^\infty(R^+)$ is a nonnegative non- increase function with $\eta(u)=1$ for $u\in[0,1]$ and $\eta(u)=0$ for $u>2$. Then $\eta_N(f,x)=\ f(x),\ f\in L_p(dw)$. Combing (9) with [18, Proposition 2.1], we have

$$\|\eta_N(f)\|_{L_p(dw)}\le C_2\|f\|_{L_p(dw)}. \qquad (9)$$

For a given $N>1$ we define the polynomial kernels by

$$K_N(x,y)=\sum_{k=0}^{2N}p_k(x)p_k(y),\ x,y\in[-1,1]. \qquad (10)$$

For a given discrete set $\bar{t}_N\subset[-1,1]$, the corresponding RKHS $\mathcal{H}_{K_N}^{\bar{t}_N}$ is defined as (Ref. e.g.[19]) the linear space of the set of functions $\{K_x:=K_N(x,t):t\in\bar{t}_N\},x\in X$ with the inner product given by

$$\langle f,g\rangle_{K_N}=\sum_{i,j}c_i d_j K_N(x_i,y_j),\ f=\sum_i c_i K_{x_i},\ g=\sum_j d_j K_{y_j}. \tag{11}$$

We assume the distribution $\rho(x,y)=\rho(y\mid x)w(x)$ with $w(x)$ being Jacobi weight. Corresponds to the scheme (3) we have the following scheme,

$$f_{z,\lambda}:=\arg\min_{f\in\mathcal{H}_{K_N}^{\bar{t}_N}}\left\{\mathcal{E}_z(f)+\lambda\| f\|^2_{K_N,\bar{t}_N}\right\}, \tag{12}$$

We define the projection operator π on the space of measurable function $f:X\to R$ as

$$\pi(f)(x)=\begin{cases}1, & \text{if} f(x)>1,\\ -1, & \text{if} f(x)<-1,\\ f(x), & \text{if } -1\le f(x)\le 1.\end{cases}$$

Since $V(y\pi(f)(x))\le V(yf(x))$, we know that $\mathcal{E}(\pi(f))\le\mathcal{E}(f),\mathcal{E}_z(\pi(f))\le\mathcal{E}_z(f)$. By virtue of $\pi(f)$, we take $\pi(f_{z,\lambda})$ instead of $f_{z,\lambda}$ to analyze the related learning rates.

Theorem 1.1. Let $\zeta=\dfrac{1}{2+p\beta(2-\alpha_p)}$, $\lambda=\exp\{-2m^{\zeta}\}$, $\alpha_p=\min\{p/2,2/p\}$ for $p>1$ and $\alpha\in[0,1]$ for $p=1$, $\beta>0$, $N\sim m^{\zeta}$ be a given number. Then, for all $m\ge(4/\zeta^2+4\log(pM^{p-1}))^{1/\zeta}$ and $0<\delta<1$, with confidence at least $1-2\delta$, we have

$$\mathcal{E}(\pi(f_{z,\lambda}))-\mathcal{E}(f_\rho^V)\le\tilde{c}\cdot m^{-\theta},\quad\theta=\frac{p\beta}{2+p\beta(2-\alpha_p)}.$$

Theorem 1.2. Under the assumption of Theorem 2.1, we have for all $0<\delta<1/2$, with confidence at least $1-2\delta$,

$$\mathcal{R}(\mathrm{sgn}(f_{z,\lambda}))-\mathcal{R}(f_c)\le\begin{cases}\tilde{c}m^{-\theta}, & \text{for p=1,}\\ c^*m^{-\theta/2}, & \text{for p>1,}\end{cases}$$

where $\theta=\dfrac{p\beta}{2+p\beta(2-\alpha_p)}$, $c^*=\sqrt{2\tilde{c}}$, and $\tilde{c}$ is the same as in Theorem 2.1.

II. THE APPROXIMATION ERROR

We estimate the approximation error for $\eta_N(f_\rho^V)$, i.e. $\mathcal{E}(\eta_N(f_\rho^V))-\mathcal{E}(f_\rho^V)+\lambda\|\eta_N(f_\rho^V)\|^2_{K_N,\bar{t}_N}$.

Let $\mathcal{P}_n$ be the set of all algebraic polynomials of order not exceeding n. For $p_n\in\mathcal{P}_n$, denoted by $\bar{t}_n=\{x_{k,n}\}_{k=1}^n$ the zeroes of $p_n(x)$ arranged as

$$-1<x_{n,n}<x_{n-1,n}<\cdots<x_{2,n}<x_{1,n}<1.$$

Then, there holds the Gauss quadrature formula (Ref. e.g.[15])

$$\int_{-1}^1 p_k(x)w(x)dx=\sum_{k=1}^n\lambda_{k,n}p_k(x_{k,n}),\ \ p_k\in\mathcal{P}_{2n-1}. \tag{13}$$

For any $p_n\in\mathcal{P}_n$, there holds the Nikolskii inequality (see, e.g.[13],[16])

$$\| p_n\|_{L_2(dw)}\le C_1 n^{(\frac{1}{2}-\frac{1}{p})_+}\| p_n\|_{L_p(dw)}, \tag{14}$$

where $a_+=\max(a,0)$, $1\le p\le\infty$.

Proposition 2.1. Let $\eta_N(f)$, $K_N(x,y)$ be defined as (10) and (12) respectively. Then, for any $f\in L_p(dw)$ $p\ge1$, $\beta>0$,

$$\mathcal{E}(\eta_N(f_\rho^V))-\mathcal{E}(f_\rho^V)+\lambda\|\eta_N(f_\rho^V)\|^2_{K_N,\bar{t}_N}\le\frac{M_3}{N^{p\beta}}+\lambda C_1^2N^{2(\frac{1}{2}-\frac{1}{p})_+}, \tag{15}$$

To prove Proposition 2.1, we firstly bound $\|\eta_N(f_\rho^V)\|_{K_N,\bar{t}_N}$.

Lemma 2.2. For any $f\in L_p(dw)$ there is $C_1>0$ such that

$$\|\eta_N(f)\|^2_{K_N,\bar{t}_N}\le C_1^2N^{2(\frac{1}{2}-\frac{1}{p})_+}\| f\|^2_{L_p(dw)}. \tag{16}$$

Lemma 2.3 ([19], Proposition 2.1). Let $D(\beta)f$ be given by (8). There is a constant $M_0>0$ such that

$$\|\eta_N(f)-f\|_{L_p(dw)}\le\frac{M_0\| D(\beta)f\|_{L_p(dw)}}{N^{\beta}}, \tag{17}$$

Lemma 2.4 ([3], Theorem 25). If $f:X\to R$ is measurable, then

$$\mathcal{E}(f)-\mathcal{E}(f_\rho^V)\le\begin{cases}\| f-f_\rho^V\|^p_{L_p(dw)}, & \text{if } 1\le p\le 2,\\ p2^{p-1}\| f-f_\rho^V\|_{L_p(dw)}(2^{p-1}+\| f-f_\rho^V\|^{p-1}_{L_p(dw)}), & \text{if } p>2.\end{cases} \tag{18}$$

Lemma 2.5. There are constants $M_1>0$, $M_2>0$, $M_3>0$ such that $p\ge1$, $\beta>0$,

$$\mathcal{E}(\eta_N(f_\rho^V))-\mathcal{E}(f_\rho^V)\le\frac{M_3}{N^{p\beta}} \tag{19}$$

where $M_3=p2^{p-1}(M_2^{p-1}+1)M_1^p$, $M_2\ge\dfrac{2N^{\beta}}{M_1}$.

Now Proposition 2.1. can be derived from Lemma 2.2 and Lemma 2.5.

III. SAMPLE ERROR

Proposition 3.1. Let $0<\zeta<\dfrac{1}{2}$, $\lambda=\exp\{-2m^{\zeta}\}$, $\alpha_p=\min\{p/2,2/p\}$ for $p>1$ and $\alpha\in[0,1]$ for $p=1$, $\beta>0$, $N\le m^{\zeta}$. Then, for all $m\ge(4/\zeta^2+4\log(pM^{p-1}))^{1/\zeta}$ and $0<\delta<1$, with confidence at least $1-2\delta$, there holds

$$\begin{aligned}&[\mathcal{E}(\pi(f_{z,\lambda}))-\mathcal{E}(f_\rho^V)]-[\mathcal{E}_z(\pi(f_{z,\lambda}))-\mathcal{E}_z(f_\rho^V)]\\ &+[\mathcal{E}_z(f)-\mathcal{E}_z(f_\rho^V)]-[\mathcal{E}(f)-\mathcal{E}(f_\rho^V)]\\ &\le\frac{1}{2}\{\mathcal{E}(\pi(f_{z,\lambda}))-\mathcal{E}(f_\rho^V)\}+c(m),\end{aligned} \tag{20}$$

where

$$c(m)=\frac{M_3}{2N^{p\beta}}+\frac{2^{p+2}\log\frac{1}{\delta}}{3m}$$
$$+\Big(\frac{2B_p\log\frac{1}{\delta}}{m}\Big)^{\frac{1}{2-\alpha_p}}+20\Big(\frac{2^{p+4}C}{3m^{1-2\zeta}}+$$
$$\frac{2^{p+3}\log\frac{1}{\delta}}{3m}+\Big(\frac{8CB_p}{m^{1-2\zeta}}+\frac{4B_p\log\frac{1}{\delta}}{m}\Big)^{1/(2-\alpha_p)}.\Big)$$

For simplicity, we divide the sample error in (13) for $f=\eta_N(f_\rho^V)$ into two terms. The first term is $A=[\mathcal{E}_z(\eta_N(f_\rho^V))-\mathcal{E}_z(f_\rho^V)]-[\mathcal{E}(\eta_N(f_\rho^V))-\mathcal{E}(f_\rho^V)]$, the second one is $B=[\mathcal{E}(\pi(f_{z,\lambda}))-\mathcal{E}(f_\rho^V)]-[\mathcal{E}_z(\pi(f_{z,\lambda}))-\mathcal{E}_z(f_\rho^V)]$. For a compact $F\subset X$, we denote by $\widetilde{N}(F,X)$ (see[23]) the set of functions in X that have more than a unique best approximation in F with respect to the norm, and $\lambda^*(f)=\sup\{\lambda'\ge 1:\lambda' f_c+(1-\lambda')f\notin\overline{\widetilde{N}(F,X)}\}$.

Lemma 3.2 ([23]). Let $1<p<\infty$ and set $F\subset L_p(dw)$ be a compact set of functions that are bounded by 1 put Y to be a random variable bounded by 1. If $Y\notin\overline{\widetilde{N}(F,L_p(dw))}$, set $\lambda^*(Y)=\sup\{\lambda'\ge 1:\lambda' f_c+(1-\lambda')Y\notin\overline{\widetilde{N}(F,X)}\}$. Then, for every $f\in F$,

$$E\{V(yf(x))-V(yf_\rho^V(x))\}^2\le B_p\{\mathcal{E}(f)-\mathcal{E}(f_\rho^V)\}^{\alpha_p},\tag{21}$$

where $B_p=c(p)\inf\limits_{1<\lambda'<\lambda^*(Y)}\frac{\lambda'}{\lambda'-1}$, $c(p)$.

Lemma 3.3. For any $0<\delta<1$, with confidence at least $1-\delta$, there holds

$$A\le\frac{2^{p+2}\log\frac{1}{\delta}}{3m}+\Big(\frac{2B_p\log\frac{1}{\delta}}{m}\Big)^{\frac{1}{2-\alpha_p}}$$
$$+\frac{1}{2}\big(\mathcal{E}(\eta_N(f_\rho^V))-\mathcal{E}(f_\rho^V)\big).\tag{22}$$

Lemma 3.4 ([1],[6]). Let ξ be a random variable on probability space Z with mean $E\xi=\mu$ and variance $\sigma^2(\xi)=\sigma^2$. If $|\xi-\mu|\le B$ almost everywhere, then for all $\tau>0$,

$$\mathrm{Prob}_{z\in\mathbb{Z}^m}\Big\{\frac{1}{m}\sum_{i=1}^m\xi(z_i)-\mu\ge\tau\Big\}\le\exp\Big\{-\frac{m\tau^2}{2(\sigma^2+\frac{1}{3}B\tau)}\Big\}.\tag{23}$$

To prove the result of sample error we need to extend (22) to a function set by means of the covering number. So let us recall some definitions. Let $\mathcal{F}$ be a subset of a metric space and $r'>0$. The covering number $\mathcal{N}(\mathcal{F},r')$ is defined to be the minimal integer $l\in\mathrm{N}$ such that there exist l balls with radius r' covering $\mathcal{F}$. Denote the covering number of the unit ball $\mathcal{B}_1$ as $\mathcal{N}(r'):=\mathcal{N}(\mathcal{B}_1,r')$, $\forall r'>0$. Take

$$\mathcal{B}_R^*=\{f\in\mathcal{H}_{K_N}^{\bar{t}_N}:\|f\|_{K_N,\bar{t}_N}\le R\}.$$

We recall some well-known results which will be used in our estimate for sample error.

Lemma 3.5 ([1]). Let E be a finite dimension Banach space with norm $\|\cdot\|_E$, $r=\dim E$, $\mathcal{B}_R=\{f\in E:\|f\|_E\le\widetilde{R}\}$, then

$$\log\mathcal{N}(\mathcal{B}_R,r')\le r\log(\frac{4\widetilde{R}}{r'}).$$

The reproducing property implies $\|f\|_\infty\le\sqrt{\sup\limits_{x\in X}K_N(x,x)}\|f\|_{K_N,\bar{t}_N}\le\sqrt{2N+1}R$, so Lemma 3.5 implies

$$\log\mathcal{N}(\mathcal{B}_R^*,r')\le(2N+1)\log\frac{4\sqrt{2N+1}R}{r'}$$
$$\le CN\log\frac{\sqrt{N}R}{r'}.\tag{24}$$

The next known result we need is a quantitative version of the law of the large numbers

Lemma 3.6 ([6]). Let $0\le\alpha\le 1$, $B>0$, $c\ge 0$, and $\mathcal{G}$ be a set of functions on Z, such that for every $g\in\mathcal{G}$, $Eg\ge 0$, $|Eg-g|\le B$ and $Eg^2\le c(Eg)^\alpha$. Then for any $\tau>0$,

$$\mathrm{Prob}_{z\in\mathbb{Z}^m}\Big\{\sup_{g\in\mathcal{G}}\frac{Eg-\frac{1}{m}\sum_{i=1}^m g(z_i)}{\sqrt{(Eg)^\alpha+\tau^\alpha}}>4\tau^{1-\frac{\alpha}{2}}\Big\}$$
$$\le\mathcal{N}(\mathcal{G},\tau)\exp\Big\{\frac{-m\tau^{2-\alpha}}{2(c+\frac{1}{3}B\tau^{1-\alpha})}\Big\}.\tag{25}$$

By direct calculation it is not difficult to derive the following lemma.

Lemma 3.7. Let $0<\zeta<\frac{1}{2}$. Then, for all $m\ge(4/\zeta^2+4\log(pM^{p-1}))^{1/\zeta}$ and $0<\delta<1$, with confidence at least $1-\delta$, we have

$$[\mathcal{E}(\pi(f_{z,\lambda}))-\mathcal{E}(f_\rho^V)]-[\mathcal{E}_z(\pi(f_{z,\lambda}))-\mathcal{E}_z(f_\rho^V)]$$
$$\le\frac{1}{2}\{\mathcal{E}(\pi(f_{z,\lambda}))-\mathcal{E}(f_\rho^V)\}+20\tau\tag{26}$$

for all $f\in\mathcal{B}_R^*$, where τ is equal to

$$\frac{2^{p+4}C}{3m^{1-2\zeta}}+\frac{2^{p+3}\log(1/\delta)}{3m}+$$
$$\Big(\frac{8CB_p}{m^{1-2\zeta}}+\frac{4B_p\log(1/\delta)}{m}\Big)^{1/(2-\alpha_p)}.\tag{27}$$

Proof of Proposition 3.1. The result of Proposition 3.1 can be derived from Lemma 3.3 and Lemma 3.6.

Ⅳ. ESTIMATE OF LEARNING RATES

Now we are in a position to present the rate of learning error. The desired learning rate can be derived by

combining the previous results on approximation error and sample error.

Proof of Theorem 1.1. Putting (17) and (22) into (16) with $f = \eta_N(f_\rho^V)$, we have

$$\mathcal{E}(\pi(f_{z,\lambda})) - \mathcal{E}(f_\rho^V) \le$$

$$\lambda C_1^2 N^{2(\frac{1}{2}-\frac{1}{p})_+} + \frac{M_3}{N^{p\beta}} + \frac{1}{2}\{\mathcal{E}(\pi(f_{z,\lambda})) - \mathcal{E}(f_\rho^V)\} + c(m),$$

where $c(m)$ given by (22), we have with confidence at least $1-2\delta$, that

$$\mathcal{E}(\pi(f_{z,\lambda})) - \mathcal{E}(f_\rho^V) \le 2\lambda C_1^2 N^{2(\frac{1}{2}-\frac{1}{p})_+}$$

$$+\frac{3M_3}{N^{p\beta}} + \frac{2^{p+3}\log\frac{1}{\delta}}{3m} + 2\Big(\frac{2B_p\log\frac{1}{\delta}}{m}\Big)^{\frac{1}{2-\alpha_p}}$$

$$+40\Big(\frac{2^{p+4}C}{3m^{1-2\zeta}} + \frac{2^{p+3}\log\frac{1}{\delta}}{3m} + \Big(\frac{8CB_p}{m^{1-2\zeta}} + \frac{4B_p\log\frac{1}{\delta}}{m}\Big)^{\frac{1}{2-\alpha_p}}\Big). \quad (28)$$

Since $N \sim m^\zeta$ and $\lambda = \exp\{-2m^\zeta\}$, we have $\lambda N^{2(\frac{1}{2}-\frac{1}{p})_+} \le \frac{1}{\exp\{m^\zeta\}}$, then (28) can be rewritten as

$$\mathcal{E}(\pi(f_{z,\lambda})) - \mathcal{E}(f_\rho^V) \le$$

$$2C_1^2\Big(\frac{1}{\exp\{m^\zeta\}}\Big) + 3M_3'\Big(\frac{1}{m^{p\beta\zeta}}\Big) + \frac{2^{p+3}\log\frac{1}{\delta}}{3m} +$$

$$\Big\{2(2B_p\log\frac{1}{\delta})^{\frac{1}{2-\alpha_p}} + 40\Big(4\log\frac{1}{\delta}\Big(B_p + \Big(\frac{2^{p+1}}{3}\Big)^{2-\alpha_p}\Big)\Big)^{\frac{1}{2-\alpha_p}}\Big\}\Big(\frac{1}{m^{2-\alpha_p}}\Big)$$

$$+40\Big(2^{p+4}C + 2^{p+3}\log\frac{1}{\delta} +$$

$$\Big(8CB_p + 4B_p\log\frac{1}{\delta}\Big)^{\frac{1}{2-\alpha_p}}\Big)\Big(m^{-\frac{1-2\zeta}{2-\alpha_p}}\Big).$$

Take $\zeta = \frac{1}{2+p\beta(2-\alpha_p)}$, we have $\mathcal{E}(\pi(f_{z,\lambda})) - \mathcal{E}(f_\rho^V) \le \tilde{c}m^{-\theta}$.

Proof of Theorem 1.2. Theorem 1.2 can be derived immediately by combining the result of Theorem 1.1 and the following relations between the expected risk and excess misclassification error:

$$\mathcal{R}(\mathrm{sgn}(f)) - \mathcal{R}(f_c) \le$$

$$\begin{cases} \mathcal{E}(\pi(f)) - \mathcal{E}(f_c), & \text{for p=1,} \\ \sqrt{2(\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho^V))}, & \text{for p}>1, \end{cases}$$

where $f: X \to R$ is measurable, Ref.[3].

Acknowledgment

This work was supported in part by a grant from Natural Science Foundation of China (Grant No.10871226, 10971251)and Zhejiang Natural SCience Foundation of Zhejiang Province(Grant No. Y6100096).

Ye Peixin is supported by the Program for New Century Excellent Talents in University of China..

REFERENCES

[1] F. Cucker, S. Smale, On the mathematical foundations of learning theory, Bull. Amer. Math. Soc. 2001, 39 (1): 1-49.
[2] D.R. Chen, Q. Wu, Y. Ying, D.X. Zhou, Support vector machine soft margin classifiers: Error analysis, J. Mach. Learn. Res. 2004, 5: 1143-1175.
[3] H. Z. Tong, D.R. Chen, L.Z. Peng, Learning rates for regularized classifiers using multivariate polynomial kernels, J. Complexity. 2008, 24: 619-631.
[4] D.X. Zhou, K. Jetter, Approximation with polynomial kernels and SVM classifiers, Adv. Comput. Math. 2006, 25: 323-344.
[5] Q. Wu, D.X. Zhou, SVM soft margin classifiers: Linear programming versus quadratic programming, Neural Comput. 2005, 17: 1160-1187.
[6] Q. Wu, D.X. Zhou, Analysis of support vector machine classification, J. Comp. Anal. Appl. 2006, 8: 99-119.
[7] Y. Lin, Support vector machined and the Bayes ruler in classification, Data Min. and Knowl. Discov. 2002, 23: 259- 275.
[8] P. G. Nevai, Orthogonal polynimials, Amer. Math. Soc. March, 1979, 18 (213).
[9] A. Cannon, J.M. Ettinger, D. Hush, C. Scovel, Machine learning with data dependent hypothesis classes, J. Mach. Learn. Res. 2002, 2: 335-358.
[10] N. Aronszajn, Theory of reproducing kernels, Trans. Amer. Math. Soc. 1950, 68 (2) : 337-404.
[11] F. Dai, Z. Ditzian, Cesaro summability and Marchaud inequality, Constr. Approx. 2007, 25: 73-88.
[12] Q. Wu, Y. Ying, D.X. Zhou , Learning rates of least-square regularized regression, Found. Comput. Math. 2006, 6: 171-192.
[13] S. Mendelson, Obtaining fast error rate in nonconvex situations, J. Complexity. 2008, 24: 380-397.
[14] D.X. Zhou, Capacity of reproducing kernel spaces in learning theory, IEEE Trans. Inf. Theory, 2003, 49(7) : 1743- 1752.
[15] F. Cucker, S. Smale, Best choices for regularization parameters in learning theory: On the bias-variance problem, Found. Compt. Math. 2002, 2: 413-428.
[16] W. Chen, Z. Ditzian, Best approximation and K-functional, Acta Math. Hungar, 1997, 75(3) : 165-208.
[17] C. A. Micchelli, M. Pontil, Learning the kernel function via regularization, J. Mach. Learn. Res., 6(2005),1099-1125.
[18] F. Dai, Z. Ditzian, Ces\`{a}ro summability and Marchaud inequality, Constr. Approx. 2007, 25: 73-88.
[19] S. Smale, D.X. Zhou, Shannon sampling, Connections to learning theory, Appl. Comput. Harmon. Anal. 2005, 19: 285-302.
[20] S. Smale, D.X. Zhou, Learning theory estimates via integral operators and their applications, Constr. Approx. 2007, 26: 153-172.
[21] Q. Wu, Y. Ying, D.X. Zhou , Learning rates of least-square regularized regression, Found. Comput. Math.}2006, 6: 171-192.
[22] S. Mendelson, Obtaining fast error rate in non-convex situations, J. Complexity. 2008, 24: 380-397.
[23] D.X. Zhou, The coveing number in learning theory, J. Complexity, 2002, 18(3) : 739-767.
[24] D.X. Zhou, Capacity of reproducing kernel spaces in learning theory, IEEE Trans. Inf. Theory}, 2003, 49(7) : 1743-1752.
[25] F. Cucker, S. Smale, Best choices for regularization parameters in learning theory: On the bias-variance problem, Found. Compt. Math. 2002, 2: 413-428.

Baohuai Sheng attended Baoji Normal College, Baoji, Shaanxi, from 1981 to 1985. He earned his BS degree in mathematical teaching from the department of mathematics in 1985. He earned his M.S. degree in basic mathematics from the department of mathematics of Hangzhou University in 1988. From 1988-2001 he worked towards the Doctor of Sciences in the applied mathematics at Xidian University, Xian, P.R.China, and earned his S. D. degree in March,2 001.From March, 2001, to September,2003, he served as a Professor at the Ningbo University. Since September,2003,he has been a faculty member of Shaoxing College of Arts and Sciences, Shaoxing, Zhejiang, P.R.China, where he serves currently as a Professor and the Chairman of Mathematical Department. His current research interests focus on the area of approximation theory, nonlinear optimization, learning theory.

Peixin Ye received the B.S. Degree and the M.S. degree in mathematics from Xiamen University, Fujian, China, in 1995 and 1998 respectively. He received the Ph.D. degree in mathematics from Beijing Normal University, in 2001. From 2001 to 2003, he worked in the Institute of mathematics , Chinese academy of sciences as a postdoctoral. He is currently a Full Professor at School of mathematical sciences, Nankai University. He has published more than 60 journal and conference papers. His current research interests include approximation theory, numerical analysis, quantum computing, machine learning and compressed sensing.

A Trust Region Algorithm Using Curve-Linear Searching Direction for Unconstrained Optimization

Shu-ping Yang
School of Mathematical Science and Computing Technology, Central South University, Changsha, 410083
yahepi@163.com

Xiu-gui Yuan, Zai-ming Liu
School of Mathematical Science and Computing Technology, Central South University, Changsha, 410083
xgyuan2000@sina.com

***Abstract*—In the paper, aimed at the shortcoming of trust region method, we proposed a algorithm using negative curvature direction as its searching direction. The convergence of the algorithm was given. Furthermore, combing trust region method and curve-linear searching techniques, a trust region algorithm, using general curve-linear searching direction, was proposed. We proved its efficiency and feasibility. The algorithm has adjustability and can select or update its searching direction according to the iteration. This allows the algorithm that has the properties of curve-linear searching method and the global convergence of trust region method. Finally, we indicate that some searching directions of common methods can be as a special searching direction of the general method.**

***Index Terms*—nonlinear programming, unconstrained optimization, trust region method, curve-linear searching method, searching direction, quadratic model, directions of negative curvature**

I. UNCONSTRAINED OPTIMIZATION PROBLEMS[1]

In many problems of unconstrained optimization such as :

$$\text{(P1)} \qquad \min_{x\in R^n} f(x)$$

Its solution has been appealed to many peoples to do it. People created many algorithms aiming at question (P1) and presented some methods as the trust region method; Newton method; DFP method and BFGS method and so on. At these methods, people using straight line as its searching direction to searching line in general. with these methods , they have some defect as "saw tooth phenomenon" and " local convergence", some people presented the trust region method and curve-linear searching techniques to quality the global convergence of iterative method Curve-linear searching method is a way of searching direction down to a curve, which will avoid certain defects compared to rectilinear direction searching[2]. In trust region method, we gain new iteration step length based on a partial model of minimized objective function on a constrained ellipsoid domain centered at the current iteration point, and the diameter of the ellipsoid is determined by the pattern of anticipation objective function of the model. Researches have been done by M. J. D. Powell [3], J. E. Dennis jr. and H. H. W. Mei[4], J. J. More [5], D. C. Sorensen[6, 7], D .C. Sorensen and J. J. More [8], G. A. Shulty. etc[9] and Yuan Y. etc[10-13]. It is needed to calculating one of the partial models of $f(x)$ at the point of x_k:

$$Q_k(w) = f_k + g_k^T w + \frac{1}{2} w^T G_k w$$

It is solved by solving its equivalence problems: It equals to:

(P2) $\min\left\{Q_k(w); \|w\| \le \Delta_k\right\}$

(P3) $\lambda \ge 0$ s.t. $G_k + \lambda I$ is a positive semi-definite matrix,
and

$$\begin{cases} (G_k + \lambda I) w = -g \\ \|w(\lambda)\| \le \Delta_k \end{cases}$$

In many studies, it is generally assumed that $G_k + \lambda I$ is a positive definite matrix, so the problem occurs when G_k is not a positive semi-definite matrix and is possibly a local minimum point. If $g_k = 0$, we can gain zero solution to question(P3)exclusively, and the iterative procedure discontinues. When G_k is not a positive semi-definite matrix, More & Sorensen[8], which means situations hard to cope with appears, indicating that G_k has one negative eigenvalue λ_1, where g_k is orthogonal to the null space of $(G_k - \lambda_1 I)$, and $\left\|(G_k - \lambda_1 I)^+ g_k\right\| < \Delta_k$ ((+)) denotes generalized inverse or pseudo-inverse), where $\lambda_1 I$ is the

doi:10.4304/jcp.7.1.258-263

largest negative eigen value. The solution is that the iterative step length be selected as:

$p_k = -(G_k - \lambda_1 I)^+ g_k + \xi_k v_k$, where v_k is eigenvector of g_k relative to λ_1 .Select ξ_k s.t $\|p_k\| = \Delta_k$

These methods above only involve single directions, so that if problems occur, the algorithm discontinues. Aiming at these problems, we present corresponding improved methods, using a curve-linear searching direction consisting of two descent directions.

II. IMPROVEMENT OF TRUST REGION ALGORITHMS

Algorithm 1

(1) Where $0 < \mu \le \eta < 1, 0 < \gamma_1 < 1 < \gamma_2, x_0, \Delta_0$

(2) Given that x_k, Δ_k, calculate g_k, G_k

(3) Decompose symmetric matrix G_k by Bunch-parlett decomposition method [13]

$G_k = L_k D_k L_k^T$

Where L_k is a triangular identity matrix $D_k = diag(d_1^{(k)}, ...d_n^{(k)})$

(4) (a) If $g_k = 0$ and $d_i^{(k)} \ge 0 (i = 1,2...n)$, let $x_* = x_k$, then stop. If not, go to the Step(b).

(b) If G_k is not a positive semi-definite matrix, and g_k is orthogonal to the null space of $G_k - \lambda_1^{(k)} I$, where $\lambda_1^{(k)}$ the largest negative eigenvalue, then go to the Step(c). If not, switch to Step (5).

(c) Use Fletcher-Freeman[14]method to determine negative curvature direction d_k。

(d) Let $x_{k+1} = x(\alpha_k) = x_k + \alpha_k d_k$, α_k is determined by the conditions below:

$$(\nabla f(x_k + \alpha d_k))^T d_k \ge \eta[\alpha d_k^T G_k d_k + g_k^T d_k]$$

$$f(x_k + \alpha d_k) \le f(x_k) + \mu\alpha g_k^T d_k + \frac{1}{2}\mu\alpha^2 d_k^T G_k d_k$$

$k := k+1;$

switch to Step (2)

(5)

i) Establish a secondary model:

$$Q_k(w) = f_k + g_k^T w + \frac{1}{2} w^T G_k w$$

ii) Find the solution of equation below:

$$p_k = \arg\min\{Q_k(w); \|w\| \le \Delta_k\}$$

iii) Calculate

$$\rho_k = \frac{ared_k}{pred_k}.$$

If $\rho_k < \mu$, then $\Delta_k := \gamma_1 \Delta_k$ and switch to Step ( ); if $\rho_k \ge \mu$, then $x_{k+1} = x_k + p_k, k := k+1$; if $\rho_k > \eta$, then $\Delta_{k+1} := \gamma_2 \Delta_k$, or else $\Delta_{k+1} := \Delta_k$ and switch to Step (4).

Theorem 1

If $f : R^n \to R$, where $x_0 \in D \subset R^n$ and $L(x_0) = \{x \in D; f(x) \le f(x_0)\}$ is a compact subset of D.There is $x_k \in L(x_0); \|G_k\| \le M$, $\{x_k\}$ is a iteration point sequence based on Algorithm 1. Then

$$\lim_{k\to\infty} g_k^T d_k = 0, \quad \lim_{k\to\infty} d_k^T G_k d_k = 0 \tag{1}$$

Prove:

If $D_1 = \{x_k; g_k = 0, g_k \perp \mathrm{Null}(G_k - \lambda_1^{(k)} I)\}$ and $\mathrm{Null}(G_k - \lambda_1^{(k)} I)$ is the null space of $G_k - \lambda_1^{(k)} I$. Let $D_2 = \{x_k\} - D_1$, and $\{x_k\}$ is an infinite point set. That if $\{x_k\}$ is finite, then based on Algorithm 1,

$$ared_k = -\frac{1}{2} w_k G(x_k + \theta w_k) w_k,$$

$$pred_k = p_k^T w_k - \frac{1}{2} w_k^T G_k w_k, \ g_k = 0, d_k = 0$$

Then conflict arises. At least one of D_1 and D_2 is infinite.

If D_1 is infinite, $L(x_0)$ is a compact set, then $\{x_k\}$ has at least one accumulation point. Let $x_k \to x_*$, based on algorithm 1:,

$$f_k - f_{k+1} \ge -\mu\left[\alpha_k g_k^T d_k + \frac{1}{2}\alpha_k^2 d_k^T G_k d_k\right]$$

and $d_k^T G_k d_k \le 0, d_k^T g_k \le 0$

$\because f(x)$ is continuous function, then

$f_k - f_{k+1} \to 0, (k \to \infty)$ so that

$\lim_{k\to\infty} \alpha_k^2 d_k^T G_k d_k = 0, \lim_{k\to\infty} \alpha_k g_k^T d_k = 0$

Let $\lim_{k\to\infty} \alpha_k = 0$, based on the continuity of $\nabla f(x_k + \alpha d_k)$ and algorithm 1, step (d):

$$1 = \lim_{k\to\infty} \frac{(\nabla f(x_k + \alpha_k d_k))^T d_k}{g_k^T d_k + \alpha_k d_k^T G_k d_k} \le \eta < 1$$

Then conflict arises. So $\alpha_k \geq \varepsilon, (k \geq k_1)$ and $\lim_{k\to\infty} d_k^T G_k d_k = 0, \lim_{k\to\infty} g_k^T d_k = 0$.

If D_2 is infinite, based on Algorithm 2: $x_k \in D_2$ is drawn by trust region algorithm, and based on the convergence of the algorithm, obviously $\lim_{k\to\infty} d_k^T G_k d_k = 0$ is supported. Then (1) is proved.

Based on the conclusion of Theorem 1 the theorem below can be proved:

Theorem 2 Under conditions of Theorem 1, and $\{\|d_k\|\}$ is limited, then $\exists c > 0$, s.t.

$$\|d_k\| \geq c\|g_k\|, x_k \to x_*, (k \to *) \text{ Then}$$

$\nabla f(x_*) = 0$ and $G(x_*)$ is positive semi-definite.

III. General searching methods

To question (P1),

let $x(\alpha) = x + \phi_1(\alpha)s + \phi_2(\alpha)d$, where $\phi_1(\alpha)$ and $\phi_2(\alpha)$ are continuous functions under condition that $\alpha \in \beta$ (where β is a compact set in R).Given that $\phi_1(0) = \phi_2(0) = 0$ (equals to $x(\alpha) = x$); s, d are descent directions of $f(x)$ at the point of x.Create a secondary model of $f(x)$ at the point of x:

$$Q(p) = f(x) + g^T p + \frac{1}{2} p^T G p \quad (2)$$

Where $p(\alpha) = x(\alpha) - x = \phi_1(\alpha)s + \phi_2(\alpha)d$ (3)

To guarantee that $Q(p) \leq f(x)$, then

$$\phi_2(\alpha) \geq 0, \alpha \in \beta\ \phi_1(\alpha) \geq 0 \quad (4)$$

s.t. Algorithm 1 is modified. To make it further, we gain a general trust region algorithm using curve-linear searching direction for unconstrained optimization:

Algorithm 2

Step1-3 are the same as (1)-(3) in Algorithm 1.

Step4 is same as (4) (a)-(c) in Algorithm 1.

(d) Let $x_{k+1} = x(\alpha_k) = x_k + p(\alpha_k)$.

α_k is determined by the conditions below:

$$\left(\nabla f\left(x_k + p(\alpha_k)\right)\right)^T d_k \geq \eta\left[\alpha d_k^T G_k d_k + g_k^T d_k\right]$$

$$f\left(x_k + p(\alpha_k)\right) \leq$$

$$f(x_k) + \mu\alpha g_k^T d_k + \frac{1}{2}\mu\alpha^2 d_k^T G_k d_k$$

$k := k+1$; switch to Step2.

Definition 1 $f^0(x)$ is defined as generalized derivative of $f(x)$ at the point of x,

$$\text{if } f^0(x) = \limsup_{t\to 0+} \frac{f(x+t) - f(x)}{t}.$$

Definition 2 ζ is defined as generalized successive derivative of $f(x)$ at the point of x , if $f^0(x) \geq \zeta$ .Note generalized successive derivative of $f(x)$ at the point of x as $\alpha_\beta f(x)$.That is

$$\alpha_\beta f(x) = \{\zeta; f^0(x) \geq \zeta\}.$$

Create the secondary model as below:

$$\begin{aligned} F(\alpha) = f(x) &+ \left(\mu_1(0) g^T s + \mu_2(0) g^T d\right)\alpha \\ &+ \frac{1}{2}\alpha^2\left[\left(\nu_1(0) g^T s + \nu_2(0) g^T d\right]\right. \\ &+ \frac{1}{2}\alpha^2\left[\left(\mu_1(0) s + \mu_2(0) d\right)^T\right] \\ &\times\left[G\left(\mu_1(0) s + \mu_2(0) d\right)\right] \end{aligned} \quad (5)$$

Obvious (5) is a generalized pattern of (2). To satisfy that $F(\alpha) \leq f(x)$, then

$$\mu_i(0) \geq 0, \nu_i(0) \geq 0, (i = 1,2)$$

Let

$\mu_i(\alpha) = \phi_i^0(\alpha), \nu_i(\alpha) = \nu_i^0(\alpha), (i = 1,2)$, then:

$$\mu_1(0) = \lim_{t\to 0+} \frac{\phi_1(t)}{t} \ , \ \mu_2 = \lim_{t\to 0+} \frac{\phi_2(t)}{t}$$

$$(i = 1,2), \zeta \in \phi(\alpha)$$

Theorem 3

Assume that $f(x)$ is a second order continuous differentiable function on R^n , $\phi_1(\alpha), \phi_2(\alpha)$ are positive continuous functions on β (a compact subset on R) $\to R$, $g(\alpha) \in \alpha_\beta f(x(\alpha))$ is continuous from the right of point x .If:

$$\mu \in (0,1), \mu_i(0) \geq 0, \nu_i(0) \geq 0, (I = 1,2),$$

if $\mu_1(0) = \mu_2(0) = 0$, and at least one of $\nu_1(0), \nu_2(0)$ is not zero, and $\|d\|, \|s\|$ are limited. And:

$$\lim_{\alpha\to 0+} \frac{\phi_1(\alpha)}{\alpha^2} \geq \mu\nu_1(0), \lim_{\alpha\to 0+} \frac{\phi_2(\alpha)}{\alpha^2} \geq \mu\nu_2(0).$$

Then $\exists \bar{\alpha} \in \beta$ to satisfy the condition that:

$$f(x(\alpha)) \leq f(x) + \mu\left[F(\alpha) - f(x)\right], \alpha \in (0, \bar{\alpha}) \quad (6)$$

Prove:

Let $\mu_1(0), \mu_2(0) \neq 0$, $\because g(\alpha) \in \alpha_\beta f(x(\alpha))$

Based on the definition:

$$f(x) - f(x(\alpha)) \geq g^T(\alpha)(x - x(\alpha))$$

$$f(x(\alpha))-f(x)\le\phi_1(\alpha)g^T(\alpha)s+\phi_2(0)g^Td$$

$$Q[\alpha]=f(x(\alpha))-f(x)-\mu$$

$$\left[F(\alpha)-f(x)\le\phi_1(\alpha)g^Ts+\phi_2(\alpha)g^Td-\mu[F(\alpha)-f(x)]\right.$$

Then

$$\lim_{\alpha\to0+}\frac{Q(\alpha)}{\alpha}\le\lim_{\alpha\to0+}\frac{\phi_1(\alpha)}{\alpha}g^TS+\ldots$$

$$\lim_{\alpha\to0+}\frac{\phi_2(\alpha)}{\alpha}g^Td-\mu\lim\frac{F(\alpha)-f(x)}{\alpha}$$

$$=(1-\mu)\left[\mu_1(0)g^Ts+\mu_2(0)g^Td\right]<0$$

If $\mu_1(0)=\mu_2(0)=0$, and at least one of $\nu_1(0),\nu_2(0)$ is not zero.

$\because g(\alpha)$ is continuous from the right of point x , that is $g(\alpha)\to g(\alpha\to0+)$, and $\|d\|,\|s\|$ are limited

$$\therefore g^T(\alpha)s\to g^Ts\quad,\quad g^T(\alpha)d\to g^Td\quad(\alpha\to0+)$$

$\therefore$ If $\varepsilon>0$, $\zeta>0$;

if $\alpha<\zeta$, then $g^T(\alpha)s\le g^Ts+\varepsilon$,

$$g^T(\alpha)d\le g^Td+\varepsilon$$

$$\therefore Q(\alpha)=\varphi_1(\alpha)g^T(\alpha)d+\varphi_2(\alpha)g^T(\alpha)s$$

$$+\left(\varphi_1(\alpha)+\varphi_2(\alpha)\right)\varepsilon-\mu\left[F(\alpha)-f(x)\right]$$

And $g^Ts\le0$, $g^Td\le0$, $\phi_1(\alpha)\ge0$, $\phi_2(\alpha)\ge0$ $(\alpha\in\beta)$

$$\therefore\lim_{\alpha\to0+}\frac{Q(\alpha)}{\alpha^2}\le\lim_{\alpha\to0+}\frac{\phi_1(\alpha)}{\alpha^2}$$

$$\left(-\mu\nu_1(0)g^TS+\left(\lim_{\alpha\to0+}\frac{\phi_2(\alpha)}{\alpha^2}-\mu\nu_1(0)g^Td\right)\right)$$

$$<0$$

Above all, $\exists\bar{\alpha}\in\beta$ to let $\alpha\in(0,\bar{\alpha}]$ and satisfy :

$$f(x(\alpha))-f(x)\le\mu[F(\alpha)-f(x)].$$

Equation (6) is proved.

Based on Theorem 3, doing linear search along the curve $p(\alpha)$ can efficiently let the value of the function decrease, and in the meantime, its flexibility allow us to build various models according to the situations during calculating, so that it can reduce the occurrence of the phenomena so-called saw tooth .

IV. THE SELECTION OF s_k AND d_k

There are two methods of selecting s_k and d_k : ①

1) Let $s_k=-g_k$, ${s_k}^Td_k\ge0$ (that is ${g_k}^Td_k\le0$)

2) Let $s_k=-g_k$, ${s_k}^T\beta_kd_k\ge0$ (that is ${g_k}^T\beta_kd_k\ge0$)

$\because$

$$ared_k=f(k)-f(x^k+{\alpha_k}^2s_k+\alpha_kd_k)$$

$$={g_k}^T({\alpha_k}^2s_k+\alpha_kd_k)+({\alpha_k}^2s_k+\alpha_kd_k)^T$$

$$\nabla^2(\bar{x}^k)({\alpha_k}^2s_k+\alpha_kd_k)$$

Where $\bar{x}^k=x^k+\theta({\alpha_k}^2s_k+\alpha_kd_k)$ $(0\le\theta\le1)$

$\therefore$

$$r_k=\frac{ared_k}{pred_k}=$$

$$\frac{{g_k}^T({\alpha_k}^2s_k+\alpha_kd_k)+({\alpha_k}^2s_k+\alpha_kd_k)^T\nabla^2f(\bar{x}^k)({\alpha_k}^2s_k+\alpha_kd_k)}{\alpha_k{g_k}^Td_k+{\alpha_k}^2({g_k}^Ts_k+\frac{1}{2}{d_k}^T\beta_kd_k)+{\alpha_k}^3{s_k}^T\beta_kd_k+\frac{1}{2}{\alpha_k}^4{s_k}^T\beta_ks_k}$$

Then:

i)where ${g_k}^Td_k\prec0$, $\lim_{\alpha_k\to0}r_k=1$

ii)where ${g_k}^Td_k=0,g_k\ne0,s_k=-g_k$,

$$\lim_{\alpha_k\to0}r_k=$$

$$\lim_{\alpha_k\to0}\frac{{\alpha_k}^2{g_k}^T+\frac{1}{2}({\alpha_k}^2s_k+\alpha_kd_k)\nabla^2f(\bar{x}^k)({\alpha_k}^2s_k+\alpha_kd_k)}{{\alpha_k}^2({g_k}^Ts_k+\frac{1}{2}{d_k}^T\beta^kd_k)+{\alpha_k}^3{s_k}^T\beta_kd_k+\frac{1}{2}{\alpha_k}^4{s_k}^T\beta_ks_k}$$

$=1$

iii)where

$$s_k=-g_k=0\ ,\ r_k=\frac{\frac{1}{2}{\alpha_k}^2{d_k}^T\nabla^2f(\bar{x}^k)d_k}{\frac{1}{2}{\alpha_k}^2{d_k}^T\beta_kd_k}$$

$$\therefore\lim_{\alpha_k\to0}r_k=1$$

So the method is trusted.

Equations relative to α :

Let $w=\alpha^2s_k+\alpha d_k$

$$\Phi(\alpha)=\alpha{g_k}^Td_k+\alpha^2({g_k}^Ts_k+\frac{1}{2}{d_k}^T\beta_kd_k)$$

$$+\alpha^3{s_k}^T\beta_kd_k+\frac{1}{2}\alpha^4{s_k}^T\beta_ks_k$$

Suppose

$$\Phi(\alpha)=\min\left\{\Phi(\alpha);\left\|\alpha^2s_k+\alpha d_k\right\|\le\Delta_k\right\}\qquad\text{(P4)}$$

Question (P4) equals to that: $\exists\gamma\ge0$, let:

$$\begin{cases}[g_k+(\beta_k+\gamma I)(\alpha d_k+\alpha^2s_k)]^T(2\alpha s_k+d_k)=0,(7.1)\\ \left\|\alpha^2s_k+\alpha d_k\right\|=\Delta_k,(7.2)\end{cases}$$

In question (P4), the constrained condition $\left\|\alpha^2s_k+\alpha d_k\right\|\le\Delta_k$ can be further strengthened to:

Assume that $\alpha\ge0$

$\because$

$$\left|\alpha^2\|s_k\|-\alpha\|d_k\|\right|\le\left\|\alpha^2s_k+\alpha d_k\right\|\le\alpha^2\|s_k\|+\alpha\|d_k\|$$

$$\therefore\alpha^2\|s_k\|+\alpha\|d_k\|\le\Delta_k\qquad(8)$$

Find the solutions of equation (8):

$$0 \le \alpha \le \frac{-\|d_k\| + \sqrt{\|d_k\|^2 + 4\Delta_k \|s_k\|}}{2\|s_k\|}$$

The range of α is:

$$0 \le \alpha \le \frac{-\|d_k\| + \sqrt{\|d_k\|^2 + 4\Delta_k \|s_k\|}}{2\|s_k\|} \quad (9)$$

② Assume that $\alpha \le 0$

$$\alpha^2 \|s_k\| - \alpha \|d_k\| \le \Delta_k \quad (8')$$

Find the solutions of equation (8'):

$$\frac{\|d_k\| - \sqrt{\|d_k\|^2 + 4\Delta_k \|s_k\|}}{2\|s_k\|} \le \alpha \le \frac{\|d_k\| + \sqrt{\|d_k\|^2 + 4\Delta_k \|s_k\|}}{2\|s_k\|}$$

The range of α is:

$$\frac{\|d_k\| - \sqrt{\|d_k\|^2 + 4\Delta_k \|s_k\|}}{2\|s_k\|} \le \alpha \le 0 \quad (9')$$

According to the above, the range of α is:

$$0 \le |\alpha| \le \frac{-\|d_k\| + \sqrt{\|d_k\|^2 + 4\Delta_k \|s_k\|}}{2\|s_k\|}$$

(7.1) can be transformed into

$$g_k{}^T d_k + \alpha[2g_k{}^T s_k + \gamma d_k{}^T d_k + d_k{}^T \beta_k d_k] + 3\alpha^2[s_k{}^T \beta_k d_k + \gamma s_k{}^T d_k] + 2\alpha^3[s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k] = 0 \quad (10)$$

Based on (10)

$$2[s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k]\alpha 3 + 3[s_k{}^T \beta_k d_k + \gamma s_k{}^T d_k]\alpha^2 + [2g_k{}^T s_k + d_k{}^T \beta_k d_k + \gamma d_k{}^T d_k]\alpha + g_k{}^T d_k = 0$$

Let

$$a = g_k{}^T d_k,$$

$$b = 2g_k{}^T s_k + d_k{}^T \beta_k d_k + \gamma d_k{}^T d_k,$$

$$c = 3[s_k{}^T \beta_k d_k + \gamma s_k{}^T d_k],$$

$$d = 2[s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k]$$

Then According to formula giving roots of cubic equations:

$$\alpha = \sqrt[3]{[\frac{1}{2d}(a - \frac{bc}{3d} + \frac{2c^3}{27d^2})] + \sqrt{[\frac{1}{2d}(a - \frac{bc}{3d} + \frac{2c^3}{27d^2})]^2 + [\frac{1}{3d}(b - \frac{c^2}{3d})]^3}} + \sqrt[3]{-\sqrt[3]{[\frac{1}{2d}(a - \frac{bc}{3d} + \frac{2c^3}{27d^2})] + \sqrt{[\frac{1}{2d}(a - \frac{bc}{3d} + \frac{2c^3}{27d^2})]^2 + [\frac{1}{3d}(b - \frac{c^2}{3d})]^3}}} = \varphi(\gamma)$$

s.t. $\Phi(\gamma) = \|\alpha^2 s_k + \alpha d_k\|$

Based on Newton iteration method, we gain the root γ_k of $\Phi(\gamma) - \Delta_k = 0$.

Then $\alpha_k = \varphi^2(\gamma_k)s_k + \varphi(\gamma_k)d_k$

If we Select s_k, d_k s.t. $s_k = -g_k, g_k{}^T d_k = 0$.

And then (7) can be transformed into

$$[2g_k{}^T s_k + \gamma d_k{}^T d_k + d_k{}^T \beta_k d_k]\alpha + 3[s_k{}^T \beta_k d_k + \gamma s_k{}^T d_k]\alpha^2 + 2[s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k] = 0$$

$\Rightarrow$ When $\alpha \ne 0$,

$$2g_k{}^T s_k + \gamma d_k{}^T d_k + d_k{}^T \beta_k d_k + 3[s_k{}^T \beta_k d_k + \gamma s_k{}^T d_k]\alpha + 2[s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k]\alpha^2 = 0$$

According to the formula of extraction of root :

$$\alpha = \frac{-3g_k{}^T \beta_k d_k}{4[s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k]} \pm \frac{\sqrt{9(S_k{}^T \beta_k d_k)^2 - 8(s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k)(2g_k{}^T s_k + \gamma d_k{}^T d_k + d_k{}^T \beta_k d_k)}}{4[s_k{}^T \beta_k s_k + \gamma s_k{}^T s_k]}$$

$$= \frac{3g_k{}^T \beta_k d_k}{4[g_k{}^T \beta_k g_k + \gamma g_k{}^T g_k]} \pm \frac{\sqrt{9(g_k{}^T \beta_k d_k)^2 - 8(g_k{}^T \beta_k g_k + \gamma g_k{}^T g_k)(-2g_k{}^T g_k + \gamma d_k{}^T d_k + d_k{}^T \beta_k d_k)}}{4[g_k{}^T \beta_k g_k + \gamma g_k{}^T g_k]}$$

$$= \frac{3g_k{}^T \beta_k d_k \pm \sqrt{9(g_k{}^T \beta_k d_k)^2 - 8g_k{}^T(\beta_k + \gamma I)g_k(-2g_k{}^T g_k + d_k{}^T(\beta_k + \gamma I)d_k}}{4g_k{}^T[\beta_k + \gamma I]g_k}$$

Among this formula, ($\pm$) is determined by (9) and (9'). And then (7.2) can be transformed into

$$\alpha^2 d_k{}^T d_k + \alpha^4 g_k{}^T g_k = \Delta_k$$

So we can select s_k, d_k conveniently to determine a_k, and to optimization searching calculate with algorithm 1or 2

ACKNOWLEDGMENT

This work was supported by Project 10971230 of the National Science Foundation.

REFERENCES

[1] Yuan Yaxiang, Sun Wenyu. Optimization theory and methods. Beijing:Science Press, 1997.

[2] J P Bulteau J P Vial, Curvilinear path and trust region in unconstrained optimization: a convergence analysis. Math. Prog. Study, 1987, 30:82-110

[3] Powell M J D. On the global convergence of trust region Algorithms for unconstrained minimization, Math. Prog, 1984, 29(3), 327-335

[4] Dennis J E. and Mei H H W. Two new unconstrained optimization algorithms which function and gradient values. J. Optim. Theory Appl. 1979, 28, 453-482

[5] More J J. Recent developments in algorithms and software for trust region method, in Mathematical Programming – the state of the art, Bachem A R, Grotschel M and Korte B eds. Springer Verlag, Berlin, 1983
[6] Sorensen D C. The trust region meths for unconstrained optimization. In Powell M J D ed. Nonlinear Optimization 1981, Academic Press, London, 1982, 29-38
[7] Sorensen D C. Newton's method with a model trust region modification. SIAM J. Numer. Anal. 1982, 19(4):409-426
[8] Sorensen D C and More J J. On the use of directions of negative curvature in a modified Newton method. Math. Prog. 1976, 16(1):1-20
[9] Shultz G A, Schnabel R B and Byrd R H. A family of trust region based algorithms for unconstrained minimization with strong gobal convergence properties. SIAM J. Numer. Anal., 1985, 22(1):47-67
[10] Yuan Y. On the convergence of trust region algorithms, Mathematica Numerica Sinica, 1996, 16:333-346
[11] Zhang Xiangsun, Chen hangwen, Zhang Juliang. A self-adaptive trust region method for unconstrained optimization, Or Transaction, 2001, 5(1):53-62
[12] Tang Jian, Duan Yurong. Trust region method for unconstrained optimization, J. of Chongqing University, 1991, 14(1): 33-41
[13] Fletcher R. Practical methods of optimization (Vol. 1, unconstrained optimization). Dundee:John Wiley & Sons Ltd., 1981
[14] Deng Naiyang. A negative curvature method and its convergence. Computing Mathematica, 1984, 6:23-28.
[15] Powell M JD. A new algorithm for unconstrained optimization, In Nolinear programming. Rosen J H, Mangasarian O L and Ritter K, eds. , New York： Academic Press , 1970
[16] Powell M J D. Convergence properties of a class of minimization algorithm", in Nonlinear programming 2, Mangasarian O L, Meyer R R and Rolinson S M, eds. , New York：Academic Press, 1975
[17] Yang s p, Yuan X G.A trust region algorithm using curve-l inear searching direction for unconstrained optimization, Hunan University academic journal 2005.

Shu-ping Yang, Hunan Province, China. Birthdate:May, 1979. is Probability and Mathematical Statistics Ph.D., graduated from Dept. Institute of Mathematical Sciences and Computing Technology Central South of University. And research interests on Digital Image Processing and the optimization theory. She is a Intermediate lecturer of Dept. Institute of Mathematical Sciences and Computing Technology Central South of University.

New Robust Stability of Uncertain Neutral-Type Neural Networks with Discrete Interval and Distributed Time-Varying Delays

Guoquan Liu
College of Automation, Chongqing University, Chongqing, China
Email: guoquanliu1982@hotmail.com

Simon X. Yang
School of Engineering, University of Guelph, Guelph, Ontario, Canada
Email: xianyi@yahoo.com

Wei Fu
College of Automation, Chongqing University, Chongqing, China
Email: linefw@163.com

***Abstract*—This paper develops a novel robust stability criterion for a class of uncertain neutral-type neural networks with discrete interval and distributed time-varying delays. By constructing a general form of Lyapunov-Krasovskii functional, using the linear matrix inequality (LMI) approach and introducing some free-weight matrices, the delay-dependent robust stability criteria are derived in terms of LMI. Number examples are given to illustrate the effectiveness of the proposed method.**

***Index Terms*—Neural networks; Robust stability; Linear matrix inequality; Neutral-type; Lyapunov-krasoviskii functional**

I. INTRODUCTION

As is well known, stability is one of the main properties of neural networks, which is a crucial feature in the design and hardware implementation of neural networks. However, in the process of information storage and transmission in neural networks, time delays as a source of oscillations, instability and other poor performance may occur. Therefore, a great number of results have been proposed to guarantee the global asymptotic or exponential stability of delayed neural networks, see [1]-[8] and references therein. Among theses, on the delay-dependent robust stability problems of neural networks with delays have received considerable attention. In addition, a special type of time delay in real systems as well as neural networks, i.e., interval time-varying delay is identified and investigated [9-16]. Interval time-varying delay is a time delay that varies in an interval in which the lower bound is not restricted to be zero. Hence, stability analysis for neural networks with interval time-varying delays has been widely investigated in recent years.

On the other hand, owing to the complicated dynamic properties of the neural cells in practice, the existing neural network models cannot characterize the properties of a neural reaction process precisely [17]. In order to describe dynamics more precisely for some complicated neural networks, a new type of the neural networks is in need to be introduced. Neural networks of this new type are called neutral neural networks or neural networks of neutral type. However, to date, the problem of robust stability analysis for neural networks of neutral type has been investigated by a few investigators [17]-[20]. In [19], the problem of global asymptotic stability for neural networks of neutral type time-varying delays has been investigated. In [20], the global exponential stability problem has been considered for a class of neutral-type impulsive neural networks with discrete and distributed delays. However, it should be pointed out that in the existing literature, parameter uncertainties and distributed time-varying delays were not taken into account in real neutral neural networks. Up to now, the robust stability analysis problem for neutral neural networks with discrete interval and distributed time-varying delays has not been full studied. Therefore, it is important and challenging to get some new stability criteria for uncertain delayed neutral-type neural networks.

Motivated by the above statements, a class of uncertain neutral-type neural networks with discrete interval and distributed time-varying delays is considered in this paper. Based on the Lyapunov-Krasovskii functional approach and the free-weight matrices technique, new robust stability criteria are developed in terms of LMIs, which can be easily calculated by MATLAB LMI toolbox. Moreover, the proposed stability criteria do not require the monotonicity of the activation functions and

Corresponding author:
E-mail address: guoquanliu1982@hotmail.com

doi:10.4304/jcp.7.1.264-271

the derivative of discrete time-varying delays being less than one, which generalize and improve those earlier methods. Finally, the validity and performance of the obtained results are illustrated by two examples.

Notations: Throughout this paper, for symmetric matrices X and Y, $X \geq Y$ (respectively, $X > Y$) means that $X - Y \geq 0$ ($X - Y > 0$) is a positive semi-definite (respectively, positive definite) matrix. The superscripts "T" and "-1" stand for matrix transposition and matrix inverse, respectively; $\Re^n$ and $\Re^{n\times n}$ denote the n-dimensional Euclidean space and the set of all $n \times n$ real matrices, respectively; $*$ represents the blocks that are readily inferred by symmetry; I denotes the unit matrix of appropriate dimensions. diag{...} denotes the block diagonal matrix. In this paper, if not explicit, matrices are assumed to have compatible dimensions.

II. Problem Formulation and Lemmas

Consider the following uncertain neutral-type neural networks with discrete interval and distributed time-varying delays:

$$\begin{aligned}\dot{y}(t) = &-(A+\Delta A(t))y(t)+(W_1+\Delta W_1(t))g(y(t))\\ &+(W_2+\Delta W_2(t))g(y(t-\tau(t))\\ &+(W_3+\Delta W_3(t))\dot{y}(t-h(t))\\ &+(W_4+\Delta W_4(t))\int_{t-r(t)}^{t} g(y(s))ds + I,\\ y(t) = &\varphi(t), \forall t \in [-\delta, 0], \delta = \max\{\tau_2, h, r\},\\ \varphi \in &\aleph([-\delta, 0], \Re^n)\end{aligned} \tag{1}$$

where $A = diang\{a_1, a_2, ..., a_n\}$ is a positive diagonal matrix. $W_i \in \Re^{n\times n} (i = 1, 2, 3, 4)$ are the interconnection weight matrices . $\Delta A(t)$ and $\Delta W_i(t)$ $(i = 1, 2, 3, 4)$ are parametric uncertainties. $\tau(t)$, $r(t)$ and $h(t)$ represent time-varying discrete, distributed and neutral delays of the system (1), respectively. $y(t) = [y_1(t), y_2(t), ..., y_n(t)]^T \in \Re^n$ is neural status vector. $g(y(t)) = [g_1(y_1(t)), g_2(y_2(t)), ..., g_n(y_n(t))]^T \in \Re^n$ is the neuron activation. $I = [I_1, I_2, ..., I_n]^T \in \Re^n$ is the constant external input vector. $\varphi(t)$ is the initial condition, where $t \in [-\delta, 0]$.

In system (1), the parameter uncertainties $\Delta A(t), \Delta W_1(t)$, $\Delta W_2(t), \Delta W_3(t)$ and $\Delta W_4(t)$ are assumed to be the following form

$$\begin{aligned}&[\Delta A(t), \Delta W_1(t), \Delta W_2(t), \Delta W_3(t), \Delta W_4(t)]\\ &= HF(t)[B_1, B_2, B_3, B_4, B_5],\end{aligned} \tag{2}$$

where H and $B_i, i = 1, 2, ..., 5$ are known real constant matrices of appropriate dimensions. The matrix $F(t)$, which may be time-varying, is unknown and satisfies $F^T(t)F(t) \leq I$.

The time-varying delays $\tau(t)$, $r(t)$ and $h(t)$ satisfy, respectively

$$\begin{aligned}&0 < \tau_1 \leq \tau(t) \leq \tau_2, \quad \dot{\tau}(t) \leq \tau_d, 0 < h(t) \leq h,\\ &h(t) \leq h_d < 1, 0 < r(t) \leq r,\end{aligned} \tag{3}$$

where $\tau_1, \tau_2, \tau_d, h, h_d, r$ are positive constants.

Throughout this paper, we assume that the neuron activation functions $g_j(\cdot), j = 1, 2, ..., n$ satisfy the following hypotheses, respectively:

Assumption.1 $g_j(\cdot)$ is bounded function for any $j = 1, 2, ..., n$.

Assumption.2 $l_i^- \leq \dfrac{g_i(x_1) - g_i(x_2)}{x_1 - x_2} \leq l_i^+, i = 1, 2, ..., n,$

where $\forall x_1, x_2 \in \Re^n, x_1 \neq x_2$. l_i^-, l_i^+ are some constants, and they can be positive, negative, and zero. So it is less restrictive than the descriptions on both the sigmond activation functions and the Lipschitz-type activation functions.

Assume $y^* = [y_1^*, y_2^*, ..., y_n^*]^T$ is an equilibrium point of (1). It can be easily derive that the transformation $x_i = y_i - y_i^*$ puts system (1) into the following form

$$\begin{aligned}\dot{x}(t) = &-(A+\Delta A(t))x(t)+(W_1+\Delta W_1(t))f(x(t))\\ &+(W_2+\Delta W_2(t))f(x(t-\tau(t)))\\ &+(W_3+\Delta W_3(t))\dot{x}(t-h(t))\\ &+(W_4+\Delta W_4(t))\int_{t-r(t)}^{t} f(x(s))ds,\\ x(t) = &\varphi(t), \forall t \in [-\delta, 0], \delta = \max\{\tau_2, h, r\},\\ \varphi \in &\aleph([-\delta, 0], \Re^n),\end{aligned} \tag{4}$$

where $x(t)$ is the state vector of the transformation system, $f(x(t)) = [f_1(x_1(t)), f_2(x_2(t)), ..., f_n(x_n(t))]^T \in \Re^n$ is the activation function. $f_i(x_i(t)) = g_i(x_i(t) + y_i^*) - g_i(y_i^*)$ with $f_i(x_i(t)) = 0$ for $i = 1, 2, ..., n$. Note that since each function $g_j(\cdot)$ satisfies the hypotheses Assumptions 1 and 2, hence $f_j(\cdot)$ satisfies

$$l_i^- \leq \frac{f_i(x_i)}{x_i} \leq l_i^+, i = 1, 2, ..., n, \forall x_i \in \Re^n, x_i \neq 0, \tag{5}$$

which implies that

$$0 \leq \frac{f_i(x_i) - l_i^- x_i}{x_i}, \quad 0 \leq \frac{l_i^+ x_i - f_i(x_i)}{x_i}, \tag{6}$$

where l_i^-, l_i^+ are some constants.

In order to obtain our main results, the following basic lemmas are introduced:

Lemma 1(Schur complement). Given constant S_1, S_2 and S_3 with appropriate dimensions, where $S_1^T = S_1$ and $S_2^T = S_2 > 0$, then $S_1 + S_3^T S_2^{-1} S_3 < 0$ if and only if

$$\begin{bmatrix} S_1 & S_3^T \\ * & -S_2 \end{bmatrix} < 0, \text{ or } \begin{bmatrix} -S_2 & S_3 \\ * & S_1 \end{bmatrix} < 0. \tag{7}$$

Lemma 2. For any real matrices X, Y and one positive definite matrix G, then following matrix inequality hold

$$X^T Y + Y^T X \leq X^T G X + Y^T G^{-1} Y. \tag{8}$$

Lemma 3. For any constant matrix $M \in \Re^{n\times n}$, $M = M^T > 0$, scalars a, b satisfying $a < b$, and vector function $\omega : [a,b] \to \Re^n$ such that the integrations concerned are well defined, then

$$\left(\int_a^b \omega(s)ds\right)^T M \left(\int_a^b \omega(s)ds\right) \le (b-a)\int_a^b \omega^T(s) M \omega(s)ds. \quad (9)$$

Lemma 4. Let $U, M, F(t)$ and N be real matrices of appropriate dimensions with U satisfying $U = U^T$, then $U + MF(t)N + N^T F^T(t) M^T < 0, \quad for\ all\ F^T(t)F(t) \le I$, if and only if there exists a scalar $\varepsilon > 0$ such that

$$U + MF(t)N + N^T F^T(t) M^T \le U + \varepsilon^{-1} MM^T + \varepsilon NN^T. \quad (10)$$

III. Main Results

A. Stability Critera for Normial System

In this section, we will perform the robust stability analysis for delayed neutral-type neural networks given as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + W_1 f(x(t)) + W_2 f(x(t-\tau(t))) \\ &\quad + W_3 \dot{x}(t-h(t)) + W_4 \int_{t-r(t)}^{t} f(x(s))ds. \end{aligned} \quad (11)$$

By constructing a Lyapunov-Krasovskill functional, we have the following theorem.

Theorem 1. For given scalars $\tau_1, \tau_2, \tau_d, h, h_d$ and r satisfy (3), the system (11) is globally asymptotically stable, if there exist matrices $P > 0, Q_i = Q_i^T > 0, i = 1,2,...,6,$ $R_i = R_i^T > 0,\ i = 1,2,3\quad C = C^T > 0,\quad U_i,\quad M_i,\ N_i,\ S_i,$ $i = 1,2,...,10$ and diagonal matrices $K > 0, T_i > 0,$ $i = 1,2$ such that the following LMI holds:

$$\Psi = \begin{bmatrix} \Psi_1 & U & M & N \\ * & \Psi_2 & 0 & 0 \\ * & * & \Psi_3 & 0 \\ * & * & * & \Psi_4 \end{bmatrix} < 0, \quad (12)$$

where

$$\Psi_1 = \Psi_1^T = \left[\Xi_{i,j}\right]_{10\times 10}, i, j = 1,2,...,10,$$
$$\Psi_2 = -\tau_1^{-1}(R_1 + R_2),$$
$$\Psi_3 = -(\tau_2 - \tau_1)^{-1}(R_2 + R_3),$$
$$\Psi_4 = -\tau_2^{-1}(R_2 + R_3), U^T = [U_i]_{1\times 10},$$
$$M^T = [M_i]_{1\times 10}, N^T = [N_i]_{1\times 10},$$
$$S^T = [S_i]_{1\times 10}, i = 1,2,...,10,$$

with

$$\begin{aligned} \Xi_{1,1} &= Q_1 + Q_3 + Q_4 + Q_5 - L_1T_1 - T_1^T L_1^T + U_1 + U_1^T \\ &\quad - S_1A - A^T S_1^T, \\ \Xi_{1,2} &= L_2T_1 + U_2^T - A^T S_2^T + S_1W_1, \\ \Xi_{1,3} &= U_3^T - M_1 + N_1 - A^T S_3^T, \\ \Xi_{1,4} &= U_4^T - A^T S_4^T, \\ \Xi_{1,5} &= U_5^T - U_1 + M_1 - A^T S_5^T, \\ \Xi_{1,6} &= U_6^T - N_1 - A^T S_6^T, \Xi_{1,7} = U_7^T - A^T S_7^T + S_1W_2, \\ \Xi_{1,8} &= P + U_8^T - A^T S_8^T - S_1, \\ \Xi_{1,9} &= U_9^T - A^T S_9^T + S_1W_3, \\ \Xi_{1,10} &= U_{10}^T - A^T S_{10}^T + + S_1W_4, \\ \Xi_{2,2} &= Q_2 + rC - T_1 - T_1^T + S_2W_1 + W_1^T S_2^T, \\ \Xi_{2,3} &= -M_2 + N_2 + W_1^T S_3^T, \\ \Xi_{2,4} &= W_1^T S_4^T, \\ \Xi_{2,5} &= -U_2 + M_2 + W_1^T S_5^T, \\ \Xi_{2,6} &= -N_2 + W_1^T S_6^T, \\ \Xi_{2,7} &= W_1^T S_7^T + S_2W_2, \\ \Xi_{2,8} &= K + W_1^T S_8^T - S_2, \\ \Xi_{2,9} &= W_1^T S_9^T + S_2W_3, \\ \Xi_{2,10} &= W_1^T S_{10}^T + S_2W_4, \\ \Xi_{3,3} &= -(1-\tau_d)Q_1 - L_1T_2 - T_2^T L_2^T - M_3 - M_3^T + N_3 + N_3^T, \\ \Xi_{3,4} &= -M_4^T + N_4^T, \\ \Xi_{3,5} &= -U_3 + M_3 - M_5^T + N_5^T, \\ \Xi_{3,6} &= -M_6^T + N_6^T - N_3, \\ \Xi_{3,7} &= L_2T_2 - M_7^T + N_7^T + S_3W_2, \\ \Xi_{3,8} &= -M_8^T + N_8^T - S_3, \\ \Xi_{3,9} &= -M_9^T + N_9^T + S_3W_3, \\ \Xi_{3,10} &= -M_{10}^T + N_{10}^T + S_3W_4, \\ \Xi_{4,4} &= -(1-h_d)Q_5, \\ \Xi_{4,5} &= -U_4 + M_4, \\ \Xi_{4,6} &= -N_4, \Xi_{4,7} = S_4W_2, \\ \Xi_{4,8} &= -S_4, \\ \Xi_{4,9} &= S_4W_3, \\ \Xi_{4,10} &= S_4W_4, \\ \Xi_{5,5} &= -Q_3 - U_5 - U_5^T + M_5 + M_5^T, \\ \Xi_{5,6} &= -U_6^T + M_6^T - N_5, \\ \Xi_{5,7} &= -U_7^T + M_7^T + S_5W_2, \\ \Xi_{5,8} &= -U_8^T + M_8^T - S_5, \\ \Xi_{5,9} &= -U_9^T + M_9^T + S_5W_3, \\ \Xi_{5,10} &= -U_{10}^T + M_{10}^T + S_5W_4, \\ \Xi_{6,6} &= -Q_4 - N_6 - N_6^T, \\ \Xi_{6,7} &= -N_7^T + S_6W_2, \\ \Xi_{6,8} &= -N_8^T - S_6, \\ \Xi_{6,9} &= -N_9^T + S_6W_3, \\ \Xi_{6,10} &= -N_{10}^T + S_6W_4, \\ \Xi_{7,7} &= -(1-\tau_d)Q_2 - T_2 - T_2^T + S_7W_2 + W_2^T S_7^T, \\ \Xi_{7,8} &= W_2^T S_8^T - S_7, \\ \Xi_{7,9} &= W_2^T S_9^T + S_7W_3, \end{aligned}$$

$$\Xi_{7,10} = W_2^T S_{10}^T + S_7 W_4,$$
$$\Xi_{8,8} = Q_6 + \tau_1 R_1 + \tau_2 R_3 + (\tau_2 - \tau_1) R_5 - S_8 - S_8^T,$$
$$\Xi_{8,9} = S_8 W_3 - S_9^T,$$
$$\Xi_{8,10} = -N_{10}^T + S_8 W_4 - S_{10}^T,$$
$$\Xi_{9,9} = -(1-h_d) Q_6 + S_9 W_3 + W_3^T S_9^T,$$
$$\Xi_{9,10} = W_3^T S_{10}^T + S_9 W_4,$$
$$\Xi_{10,10} = -r^{-1} C + S_{10} W_4 + W_4^T S_{10}^T.$$

Proof: Consider the following lyapunov-krasoskill functional for system (11) as

$$V(x(t),t) = \sum_i^5 V_i(x(t),t), \tag{13}$$

where

$$V_1(x(t),t) = x^T(t) P x(t) + 2\sum_{i=1}^n k_i \int_0^{x_i} f_i(s) ds,$$
$$V_2(x(t),t) = \int_{t-\tau(t)}^t \left[x^T(s) Q_1 x(s) + f^T(x(s)) Q_2 f(x(s)) \right] ds + \int_{t-\tau_1}^t x^T(s) Q_3 x(s) ds + \int_{t-\tau_2}^t x^T(s) Q_4 x(s) ds,$$
$$V_3(x(t),t) = \int_{t-h(t)}^t \left[x^T(s) Q_5 x(s) + Z^T(s) Q_6 Z(s) \right] ds,$$
$$V_4(x(t),t) = \int_{t-\tau_1}^t \int_\theta^t Z^T(s) R_1 Z(s) ds d\theta + \int_{t-\tau_2}^t \int_\theta^t Z^T(s) R_2 Z(s) ds d\theta + \int_{t-\tau_2}^{t-\tau_1} \int_\theta^t Z^T(s) R_3 Z(s) ds d\theta,$$
$$V_5(x(t),t) = \int_{-r}^0 \int_{t+\theta}^t f^{\mathrm{T}}(x(s)) C f(x(s)) ds d\theta,$$

with

$$Z(t) = \dot{x}(t),$$
$$Z(t) = A(t)x(t) + W_1(t) f(x(t)) + W_2(t) f(x(t-\tau(t))) + W_3(t) Z(t-h(t)) + W_4(t) \int_{t-r(t)}^t f(x(s)) ds. \tag{14}$$

Calculating the derivative of $V(x(t),t)$ along the solutions of system (11)

$$\dot{V}(x(t),t) = \dot{V}_1(x(t),t) + \dot{V}_2(x(t),t) + \dot{V}_3(x(t),t) + \dot{V}_4(x(t),t) + \dot{V}_5(x(t),t) \tag{15}$$
$$\dot{V}_1(x(t),t) = x^T(t) P Z(t) + 2 f^T(x(t)) K Z(t), \tag{16}$$
$$\begin{aligned}\dot{V}_2(x(t),t) &= x^T(t) Q_1 x(t) - (1-\dot{\tau}(t)) x^T(t-\tau(t)) Q_1 \times x(t-\tau(t)) + f^T(x(t)) Q_2 f(x(t)) \\ &\quad - (1-\dot{\tau}(t)) Q_4 f^T(x(t-\tau(t))) f(x(t-\tau(t))) + x^T(t) Q_3 x(t) - x^T(t-\tau_1) Q_3 x(t-\tau_1) \\ &\quad + x^T(t) Q_4 x(t) - x^T(t-\tau_2) Q_4 x(t-\tau_2) \\ &\le x^T(t) Q_1 x(t) - (1-\tau_d) x^T(t-\tau(t)) Q_1 \times x(t-\tau(t)) + f^T(x(t)) Q_2 f(x(t)) \\ &\quad - (1-\tau_d) Q_4 f^T(x(t-\tau(t))) f(x(t-\tau(t))) + x^T(t) Q_3 x(t) - x^T(t-\tau_1) Q_3 x(t-\tau_1) \\ &\quad + x^T(t) Q_4 x(t) - x^T(t-\tau_2) Q_4 x(t-\tau_2),\end{aligned} \tag{17}$$
$$\begin{aligned}\dot{V}_3(x(t),t) &= x^T(t) Q_5 x(t) - (1-\dot{h}(t)) x^T(t-h(t)) Q_5 \times x(t-h(t)) + \dot{x}^T(t) Q_6 \dot{x}(t) - (1-\dot{h}(t)) \times \dot{x}^T(t-h(t)) Q_6 \dot{x}(t-h(t)) \\ &\le x^T(t) Q_5 x(t) - (1-h_d) x^T(t-h(t)) Q_5 \times x(t-h(t)) + Z^T(t) Q_6 Z(t) - (1-h_d) \times Z^T(t-h(t)) Q_6 Z(t-h(t)),\end{aligned} \tag{18}$$
$$\begin{aligned}\dot{V}_4(x(t),t) &= \dot{x}^T(t) \left[\tau_1 R_1 + \tau_2 R_2 + (\tau_2 - \tau_1) R_3 \right] \dot{x}(t) \\ &\quad - \int_{t-\tau_1}^t Z^T(s) R_1 Z(s) ds \\ &\quad - \int_{t-\tau_2}^t Z^T(s) R_2 Z(s) ds \\ &\quad - \int_{t-\tau_2}^{t-\tau_1} Z^T(s) R_3 Z(s) ds,\end{aligned} \tag{19}$$
$$\begin{aligned}\dot{V}_5(x(t),t) &= r f^T(x(t)) C f(x(t)) \\ &\quad - \int_{t-r}^t f^T(x(s)) C f(x(s)) ds.\end{aligned} \tag{20}$$

By lemma 2 and (3), we have

$$\begin{aligned}&-\int_{t-\tau_1}^t Z^T(s) R_1 Z(s) ds \\ &= -\tau_1^{-1} \tau_1 \int_{t-\tau_1}^t Z^T(s) R_1 Z(s) ds \\ &\le -\tau_1^{-1} \left[\int_{t-\tau_1}^t Z(s) ds \right]^T R_1 \left[\int_{t-\tau_1}^t Z(s) ds \right],\end{aligned} \tag{21}$$
$$\begin{aligned}&-\int_{t-\tau_2}^t Z^T(s) R_2 Z(s) ds \\ &= -\int_{t-\tau_1}^t Z^T(s) R_2 Z(s) ds - \int_{t-\tau(t)}^{t-\tau_1} Z^T(s) R_2 Z(s) ds \\ &\quad - \int_{t-\tau_2}^{t-\tau(t)} Z^T(s) R_2 Z(s) ds \\ &\le -\tau_1^{-1} \left[\int_{t-\tau_1}^t Z(s) ds \right]^T R_2 \left[\int_{t-\tau_1}^t Z(s) ds \right] \\ &\quad - (\tau_2 - \tau_1)^{-1} \left[\int_{t-\tau(t)}^{t-\tau_1} Z(s) ds \right]^T R_2 \left[\int_{t-\tau(t)}^{t-\tau_1} Z(s) ds \right] \\ &\quad - \tau_2^{-1} \left[\int_{t-\tau_2}^{t-\tau(t)} Z(s) ds \right]^T R_2 \left[\int_{t-\tau_2}^{t-\tau(t)} Z(s) ds \right],\end{aligned} \tag{22}$$
$$\begin{aligned}&-\int_{t-\tau_2}^{t-\tau_1} Z^T(s) R_3 Z(s) ds \\ &= -\int_{t-\tau(t)}^{t-\tau_1} Z^T(s) R_3 Z(s) ds - \int_{t-\tau_2}^{t-\tau(t)} Z^T(s) R_3 Z(s) ds \\ &\le -(\tau_2 - \tau_1)^{-1} \left[\int_{t-\tau(t)}^{t-\tau_1} Z(s) ds \right]^T R_3 \left[\int_{t-\tau(t)}^{t-\tau_1} Z(s) ds \right] \\ &\quad - \tau_2^{-2} \left[\int_{t-\tau_2}^{t-\tau(t)} Z(s) ds \right]^T R_3 \left[\int_{t-\tau_2}^{t-\tau(t)} Z(s) ds \right],\end{aligned} \tag{23}$$
$$\begin{aligned}&-\int_{t-r}^t f^T(x(s)) C f(x(s))) ds \\ &\le -\int_{t-r(t)}^t f^T(x(s)) C f(x(s)) ds \\ &= -r^{-1}(t) r(t) \int_{t-r(t)}^t f^T(x(s)) C f(x(s)) ds \\ &\le -r^{-1} \left[\int_{t-r(t)}^t f(x(s)) ds \right]^T C \left[\int_{t-r(t)}^t f(x(s)) ds \right].\end{aligned} \tag{24}$$

From (6), we know that

$$\left[f_i(x_i(t)) - l_i^- x_i(t) \right]\left[f_i(x_i(t)) - l_i^+ x_i(t) \right] \le 0,$$
$$\left[f_i(x_i(t-\tau(t)) - l_i^- x_i(t-\tau(t)) \right]$$
$$\times\left[f_i(x_i(t-\tau(t))) - l_i^+ x_i(t-\tau(t)) \right] \le 0.$$

Then, for any $T_j = diag\{t_{1j}, t_{2j}, ..., t_{nj}\} \ge 0$, $j = 1,2$, it follows that

$$\begin{aligned} 0 &\le -2\sum_{i=1}^{n} t_{i1}\left[f_i(x_i(t)) - l_i^- x_i(t) \right]\left[f_i(x_i(t)) - l_i^+ x_i(t) \right] \\ &= -2 f^T(x(t)) T_1 f(x(t)) + 2x^T(t) L_2 T_1 f(x(t)) \\ &\quad - 2x^T(t) L_1 T_1 x(t) \end{aligned} \tag{25}$$

and

$$\begin{aligned} 0 &\le -2\sum_{i=1}^{n} t_{i2}\left[f_i(x_i(t-\tau(t)) - l_i^- x_i(t-\tau(t)) \right] \\ &\quad \times\left[f_i(x_i(t-\tau(t))) - l_i^+ x_i(t-\tau(t)) \right] \\ &= -2 f^T(x(t-\tau(t))) T_2 f(x(t-\tau(t))) \\ &\quad + 2x^T(t-\tau(t)) L_2 T_2 f(x(t-\tau(t))) \\ &\quad - 2x^T(t-\tau(t)) L_1 T_2 x(t-\tau(t)), \end{aligned} \tag{26}$$

where

$$L_1 = diag\{l_1^- l_1^+, l_2^- l_2^+, ..., l_n^- l_n^+\}$$

and

$$L_2 = diag\{l_1^- + l_1^+, l_2^- + l_2^+, ..., l_n^- + l_n^+\}.$$

From the Leibniz-Newton formula and Eq. (11), we can see that the following equations are true for any matrices U, M, N and S with appropriate dimensions.

$$2\zeta^T(t) U\left[x(t) - x(t-\tau_1) - \int_{t-\tau_1}^{t} Z(s)ds \right] = 0, \tag{27}$$

$$2\zeta^T(t) M\left[x(t-\tau_1) - x(t-\tau(t)) - \int_{t-\tau(t)}^{t-\tau_1} Z(s)ds \right] = 0, \tag{28}$$

$$2\zeta^T(t) N\left[x(t-\tau(t)) - x(t-\tau_2) - \int_{t-\tau_2}^{t-\tau(t)} Z(s)ds \right] = 0, \tag{29}$$

and

$$\begin{aligned} &2\zeta^T(t) S\left[-Ax(t) + W_1 f(x(t)) + W_2 f(x(t-\tau(t)) \right. \\ &\quad \left. + W_3 Z(t-h(t)) + W_4 \int_{t-r(t)}^{t} f(x(s))ds - Z(t) \right] = 0. \end{aligned} \tag{30}$$

where

$$\begin{aligned} \zeta^T(t) = \{ & x^T(t), f^T(x(t)), x^T(t-\tau(t)), x^T(t-h(t)), \\ & x^T(t-\tau_1), x^T(t-\tau_2), f^T(x(t-\tau(t))), Z^T(t), \\ & Z^T(t-h(t)), \left[\int_{t-r(t)}^{t} f(x(s))ds \right]^T \}. \end{aligned}$$

Now, by Lemmas 2 and 3, the following inequalities hold:

$$\begin{aligned} &-2\zeta^T(t) U \int_{t-\tau_1}^{t} Z(s)ds \\ &\le \zeta^T(t) U\left(\tau_1^{-1} R_1 + \tau_1^{-1} R_2 \right)^{-1} U^T \zeta(t) \\ &\quad + \left[\int_{t-\tau_1}^{t} Z(s)ds \right]^T \left(\tau_1^{-1} R_1 + \tau_1^{-1} R_2 \right)\left[\int_{t-\tau_1}^{t} Z(s)ds \right], \end{aligned} \tag{31}$$

$$\begin{aligned} &-2\zeta^T(t) M \int_{t-\tau(t)}^{t-\tau_1} Z(s)ds \\ &\le \zeta^T(t) M\left[(\tau_2 - \tau_1)^{-1}(R_2 + R_3) \right]^{-1} \\ &\quad \times M^T \zeta(t) + \left[\int_{t-\tau(t)}^{t-\tau_1} Z(s)ds \right]^T (\tau_2 - \tau_1)^{-1}(R_2 + R_3) \\ &\quad \times\left[\int_{t-\tau(t)}^{t-\tau_1} Z(s)ds \right], \end{aligned} \tag{32}$$

and

$$\begin{aligned} &-2\zeta^T(t) N \int_{t-\tau_2}^{t-\tau(t)} Z(s)ds \\ &\le \zeta^T(t) N\left[\tau_2^{-1}\left(R_2 + R_3 \right) \right]^{-1} N^T \zeta(t) \\ &\quad + \left[\int_{t-\tau_2}^{t-\tau(t)} Z(s)ds \right]^T \left[\tau_2^{-1}\left(R_2 + R_3 \right) \right]\left[\int_{t-\tau_2}^{t-\tau(t)} Z(s)ds \right]. \end{aligned} \tag{33}$$

Substituting (16)-(33) into (15), we can get

$$\begin{aligned} \dot{V}(x(t),t) &\le \zeta^T(t)\left\{ \Psi_1 + U\left(\tau_1^{-1} R_1 + \tau_1^{-1} R_2 \right)^{-1} U^T \right. \\ &\quad + M\left[(\tau_2 - \tau_1)^{-1}(R_2 + R_3) \right]^{-1} M^T \\ &\quad \left. + N\left[\tau_2^{-1}\left(R_2 + R_3 \right) \right]^{-1} N^T \right\} \zeta(t) \\ &= \zeta^T(t) \Psi \zeta(t), \end{aligned}$$

where

$$\begin{aligned} \Psi &= \Psi_1 + U\left(\tau_1^{-1} R_1 + \tau_1^{-1} R_3 \right)^{-1} U^T \\ &\quad + M\left[(\tau_2 - \tau_1)^{-1}(R_3 + R_5) \right]^{-1} M^T \\ &\quad + N\left[\tau_2^{-1}\left(R_3 + R_5 \right) \right]^{-1} N^T. \end{aligned} \tag{34}$$

By Lemma1 (Schur complements), we know that LMI (12) is equivalent to (34). It is obvious that for $\Psi < 0$, there exists a scalar $\gamma > 0$ such that

$$\dot{V}(x(t),t) \le -\gamma \left\| x(t) \right\|^2,$$

thus, if $\Psi < 0$, the system (11) is globally, asymptotically stable in the mean square. The proof is completed.

Remark 1. This result does not require for the derivative of the discrete time-varying delay $\tau(t)$ to be less than one, which is needed in [20]. Hence, the new result here is less conservative.

B. Robust Stability Criteria for Uncertain System

It is well known that, parameter uncertainties often appear in modeled neural networks, it is necessary and important to consider the robust stability of system (4). Based on Theorem 1, we can derive the following theorem for the robust stability of system (4) with uncertainties satisfying (2).

Theorem 2. For given scalars $\tau_1, \tau_2, \tau_d, h, h_d$ and r satisfy (3), the system (4) is globally robustly asymptotically stable, if there exist matrices $P > 0, Q_i = Q_i^T > 0$, $i = 1,2,...,6$, $R_i = R_i^T > 0$, $i = 1,2,3$, $C = C^T > 0$, U_i, M_i, N_i, S_i, $i = 1,2,...,10$ and diagonal matrices $K > 0, T_i > 0$, $i = 1,2$, a scalar $\varepsilon > 0$ such that the following LMI holds:

$$\begin{bmatrix} \tilde{\Psi}_1 & U & M & N & \Gamma \\ * & \Psi_2 & 0 & 0 & 0 \\ * & * & \Psi_3 & 0 & 0 \\ * & * & * & \Psi_4 & 0 \\ * & * & * & * & -\varepsilon I \end{bmatrix} < 0. \quad (35)$$

where

$\tilde{\Psi}_1 = \tilde{\Psi}_1^T = \left[\tilde{\Xi}_{i,j}\right]_{10\times 10}, \tilde{\Xi}_{i,j} = \Xi_{i,j}, i, j = 1,2,...,10,$

$\tilde{\Xi}_{1,1} \neq \Xi_{1,1}, \tilde{\Xi}_{2,2} \neq \Xi_{2,2}, \tilde{\Xi}_{7,7} \neq \Xi_{7,7},$

$\tilde{\Xi}_{8,8} \neq \Xi_{8,8}, \tilde{\Xi}_{10,10} \neq \Xi_{10,10},$

$\Gamma^T = \left[S_i H\right]_{1\times 10}, i = 1,2,...,10,$

with

$$\begin{aligned} \tilde{\Xi}_{1,1} &= Q_1 + Q_3 + Q_4 + Q_5 - L_1 T_1 - T_1^T L_1^T + U_1 + U_1^T \\ &\quad - S_1 A - A^T S_1^T + \varepsilon T_1^T T_1, \\ \tilde{\Xi}_{2,2} &= Q_2 + rZ - T_1 - T_1^T + S_2 W_1 + W_1^T S_2^T + \varepsilon T_2^T T_2, \\ \tilde{\Xi}_{7,7} &= -(1-\tau_d) Q_2 - T_2 - T_2^T + S_7 W_2 + W_2^T S_7^T + \varepsilon T_3^T T_3, \\ \tilde{\Xi}_{8,8} &= Q_6 + \tau_1 R_1 + \tau_2 R_3 + (\tau_2 - \tau_1) R_5 - S_8 - S_8^T + \varepsilon T_4^T T_4, \\ \tilde{\Xi}_{10,10} &= -r^{-1} Z + S_{10} W_4 + W_4^T S_{10}^T + \varepsilon T_5^T T_5. \end{aligned}$$

Proof: By Lemma 1 (Schur complement) and Eq. (5), the system (4) is globally robustly, asymptotically stable in the mean square if the following inequality holds:

$$\Psi + \Omega_1 F(t) \Omega_2 + \Omega_2^T F^T(t) \Omega_1^T < 0, \quad (36)$$

where Ψ is defined in Theorem 1,

$$\Omega_1 = \begin{bmatrix} S_1 H & S_2 H & S_3 H & S_4 H & S_5 H & S_6 H & S_7 H \\ S_8 H & S_9 H & S_{10} H & 0 & 0 & 0 \end{bmatrix}^T$$

and

$$\Omega_2 = \begin{bmatrix} -V_1 & V_2 & 0 & 0 & 0 & 0 & V_3 & V_4 & 0 & V_5 & 0 & 0 & 0 \end{bmatrix}.$$

From Lemma 4, (2) holds for all $F^T(t)F(t) \le I$, if and only if there exist a scalar ε, such that

$$\Psi + \varepsilon^{-1} \Omega_1 \Omega_1^T + \varepsilon \Omega_2^T \Omega_2 < 0. \quad (37)$$

It follows from the Schur complement that (37) is equivalent to the LMI (35). Then, if the LMI given in (35) hold, the neutral system (4) is globally robustly asymptotically stable in the mean square. The proof is completed.

Remark 2. As shown above, from Theorems 1 and 2, we can give stability criteria for neutral-type neural networks with time-varying delays, especially, the discrete delay is belong to a time-varying interval, which can include the fast and slow time-varying delays.

IV. ILLUSTRATIVE EXAMPLES

In this section, we present two numerical examples to illustrate the validity of our results.

Example 1. Consider the following two-dimensional delayed neutral-type neural network o given in (11) with parameters

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, W_1 = \begin{bmatrix} \alpha & 0.3 \\ 0.3 & 0.5 \end{bmatrix}, W_2 = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{bmatrix},$$

$$W_3 = \begin{bmatrix} 0.15 & 0.1 \\ -0.1 & 0.15 \end{bmatrix}, W_4 = \begin{bmatrix} 0.3 & 0.2 \\ 0.1 & 0.3 \end{bmatrix},$$

$l_1^- = 0.3, l_1^+ = 0.7, l_2^- = 0.3, l_2^+ = 0.7,$

where α is a positive scalar.

Let $\tau_1 = 0.5, \tau_2 = h = 2, h_d = 0.5, r = 0.3, L_1 = 0.21I, L_2 = I$, the problem is to determine the maximum allowable bound of α for guaranteeing the stability of the system (11). By solving the LMI (12) given in Theorem 1, one can obtain the maximum bound α_M of α and it is listed in Table 1 for different cases.

TABLE I. CALCULATED UPPER BOUNDS OF α FOR VARIOUS τ_d

τ_d	0	0.4	0.8	1.6
α	2.471	2.470	2.469	2.466

Example 2. Consider the following two-dimensional uncertain neutral-type neural network of with discrete and distributed time-varying delays

$$\begin{aligned} \dot{x}(t) &= -(A + \Delta A(t)) x(t) + (W_1 + \Delta W_1(t)) f(x(t)) \\ &\quad + (W_2 + \Delta W_2(t)) f(x(t - \tau(t))) \\ &\quad + (W_3 + \Delta W_3(t)) \dot{x}(t - h(t)) \\ &\quad + (W_4 + \Delta W_4(t)) \int_{t-r(t)}^{t} f(x(s)) ds, \end{aligned} \quad (38)$$

where

$$A = \begin{bmatrix} 1.2 & 0 \\ 0 & 1 \end{bmatrix}, W_1 = \begin{bmatrix} -1.2 & 0.2 \\ 0.26 & 0.1 \end{bmatrix}, W_2 = \begin{bmatrix} -0.1 & 0.2 \\ 0.2 & -0.1 \end{bmatrix},$$

$$W_3 = \begin{bmatrix} -0.2 & 0 \\ 0.2 & -0.1 \end{bmatrix}, W_4 = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.8 \end{bmatrix},$$

$B_1 = B_2 = B_3 = B_4 = B_5 = 0.2I, H = I,$

$l_1^- = 0.2, l_1^+ = 0.8, l_2^- = 0.2, l_2^+ = 0.8.$

Let $\tau_1 = 0.8$, $\tau_2 = 1.6$, $\tau_d = h_d = 0.8$, $r = 0.2$, $h = 0.9$, $L_1 = 0.26I, L_2 = I$. By applying Theorem 2, there exists a feasible solution which guarantees the globally robustly stability of the system (38). Limited the length of the paper, we only show a part of the feasible solution:

$$P = \begin{bmatrix} 39.1345 & 5.4708 \\ 5.4708 & 28.6173 \end{bmatrix}, Q_1 = \begin{bmatrix} 9.6385 & 1.0086 \\ 1.0086 & 6.4538 \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} 13.3056 & -0.2246 \\ -0.2246 & 9.0956 \end{bmatrix}, Q_3 = \begin{bmatrix} 11.0897 & 1.3563 \\ 1.3563 & 8.5016 \end{bmatrix},$$

$$Q_4 = \begin{bmatrix} 10.1669 & 1.0214 \\ 1.0214 & 10.1669 \end{bmatrix}, Q_5 = \begin{bmatrix} 14.9123 & 1.4304 \\ 1.4304 & 9.6367 \end{bmatrix},$$

$$Q_6 = \begin{bmatrix} 17.6123 & 2.0489 \\ 2.0489 & 10.2962 \end{bmatrix}, R_1 = \begin{bmatrix} 5.3173 & 1.4545 \\ 1.4545 & 6.2822 \end{bmatrix},$$

$$R_2 = \begin{bmatrix} 3.9203 & 1.1083 \\ 1.1083 & 4.6760 \end{bmatrix}, R_3 = \begin{bmatrix} 6.4508 & 1.4526 \\ 1.4526 & 7.4101 \end{bmatrix},$$

$$C = \begin{bmatrix} 10.1738 & 0.7400 \\ 0.7400 & 15.5877 \end{bmatrix},$$

$$K = \mathrm{diag}\{21.0818 \quad 21.0818\},$$

$$T_1 = \mathrm{diag}\{23.4141 \quad 23.4141\},$$

$$T_2 = \mathrm{diag}\{10.1564 \quad 10.1564\},$$

$$\varepsilon = 11.8443.$$

V. CONCLUSIONS

This paper studied the global robust stability analysis problem for uncertain neutral-type neural networks with discrete interval and distributed time-varying delays. Novel global robust stability criteria are derived based on the Lyapunov-Krasovskii function method and the free-weight matrices technique. The proposed criteria are expressed in terms of LMI, which can be easily by using the MALTAB LMI control toolbox. Finally, numerical examples are given to show the usefulness of the proposed stability results.

ACKNOWLEDGMENT

This work is supported by the Fundamental Research Funds for the Central Universities (No. CDJXS11172237).

REFERENCES

[1] S. Arik, “An analysis of global asymptotic stability of delayed cellular neural networks,” IEEE Trans Neural Network, vol.13, no.5, pp. 1239-1242, Sep 2002.
[2] T. Amemiya and W. B. Ma, “Global asymptotic stability of nonlinear delayed systems of neutral type,” Nonlinear Anal-Theor, vol.54, no.1, pp. 83-91, Jul 2003.
[3] V. Singh, “A generalized lmi-based approach to the global asymptotic stability of delayed cellular neural networks,” IEEE Trans Neural Network, vol.15, no.1, pp. 223-225, Jan 2004.
[4] D. G. Yang and C. Y. Hu, “Novel delay-dependent global asymptotic stability condition of hopfield neural networks with delays,” Comput Math Appl, vol.57, no.11-12, pp. 1978-1984, Jun 2009.
[5] Y. J. Shen, H. Yu and J. G. Jian, “Delay-dependent global asymptotic stability for delayed cellular neural networks,” Commun Nonlinear Sci, vol.14, no.4, pp. 1057-1063, Apr 2009.
[6] Q. H. Zhou and L. Wan, “Exponential stability of stochastic delayed hopfield neural networks,” Appl Math Comput, vol.199, no.1, pp. 84-89, 2008.
[7] B. Y. Zhang, S. Y. Xu and Y. M. Li, “Delay-dependent robust exponential stability for uncertain recurrent neural networks with time-varying delays,” Int J Neural Syst, vol.17, no.3, pp. 207-218, 2007.
[8] K. Y. Liu and H. Q. Zhang, “An improved global exponential stability criterion for delayed neural networks,” Nonlinear Anal-Real, vol.10, no.4, pp. 2613-2619, Aug 2009.
[9] C. Peng and Y. C. Tian, “Delay-dependent robust stability criteria for uncertain systems with interval time-varying delay,” J Comput Appl Math, vol.214, no.2, pp. 480-494, 2008.
[10] O. M. Kwon and J. H. Park, “Delay-range-dependent stabilization of uncertain dynamic systems with interval time-varying delays,” Appl Math Comput, vol.208, no.1, pp. 58-68, 2009.
[11] S. H. Kim and P. Park, “Relaxed h-infinity stabilization conditions for discrete-time fuzzy systems with interval time-varying delays,” IEEE Trans on Fuzzy Systems, vol.17, no.6, pp. 1441-1449, 2009.
[12] K. W. Yu and C. H. Lien, “Stability criteria for uncertain neutral systems with interval time-varying delays,” Chaos Soliton Fract, vol.38, no.3, pp. 650-657, Nov 2008.
[13] Y. Y. Hou, T. L. Liao, C. H. Lien and J. J. Yan, “Stability analysis of neural networks with interval time-varying delays,” Chaos, vol.17, no.3, pp. 033120, Sep 2007.
[14] O. M. Kwon, J. H. Park and S. M. Lee, “On robust stability for uncertain neural networks with interval time-varying delays,” Iet Control Theory and Applications, vol.2, no.7, pp. 625-634, Jul 2008.
[15] O. M. Kwon and J. H. Park, “Exponential stability analysis for uncertain neural networks with interval time-varying delays,” Appl Math Comput, vol.212, no.2, pp. 530-541, Jun 15 2009.
[16] J. Tian and X. Zhou, “Improved asymptotic stability criteria for neural networks with interval time-varying delay,” Expert Syst Appl, vol.37, no.12, pp. 7521-7525, 2010.
[17] J. H. Park, O. M. Kwon and S. M. Lee, “Lmi optimization approach on stability for delayed neural networks of neutral-type,” Appl Math Comput, vol.196, no.1, pp. 236-244, Feb 15 2008.
[18] G. Wang, W. Chen and Y. Liu, “Delay-dependent global robust stability criterion for neural networks with neutral-type time-varying delays,” Proceedings of the Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering, vol.224, no.I4, pp. 321-327 , 2010.
[19] J. H. Park and O. M. Kwon, “Global stability for neural networks of neutral-type with interval time-varying delays,” Chaos Soliton Fract, vol.41, no.3, pp. 1174-1181, Aug 15 2009.
[20] R. Samidurai, S. Marshal Anthoni and K. Balachandran, “Global exponential stability of neutral-type impulsive neural networks with discrete and distributed delays,” Nonlinear Analysis: Hybrid Systems, vol.4, no.1, pp. 103-112, 2010.

Guoquan Liu was born in 1982. He received the B.Sc. degree in electronic information engineering from Zhengzhou University, Zhengzhou, China, in 2005, and his M.Sc. degree in pattern recognition and intelligent system from Sichuan University of Science and Engineering, Zigong, China, in 2008. He is now pursuing his Ph.D. degree in College of Automation, Chongqing University, China. His research interests include nonlinear systems, neural networks, and stochastic stability analysis.

Simon X. Yang received the B.Sc. degree in Engineering Physics from Beijing University, China, in 1987, the first M.Sc. degree in Biophysics from Chinese Academy of Sciences, Beijing, China, in 1990, the second M.Sc. degree in Electrical Engineering from the University of Houston, USA, in 1996, and the Ph.D. degree in Electrical and Computer Engineering from the University of Alberta, Edmonton, Canada, in 1999. He joined the University of Guelph in Canada in August 1999 right

after his Ph.D. graduation. Currently he is a Professor and the Head of the Advanced Robotics and Intelligent Systems (ARIS) Laboratory at the University of Guelph. Prof. Yang's research expertise is in the areas of Robotics, Intelligent Systems, Control Systems, Sensing and Multi-sensor Fusion, and Computational Neuroscience. Dr. Yang has served as an Associate Editor of IEEE Transactions on Neural Networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B, International Journal of Robotics and Automation, and has served as the Associate Editor or Editorial Board Member of several other international journals. He has involved in the organization of many international conferences. He was a recipient of the Presidential Distinguished Professor Awards at the University of Guelph, Canada.

Wei Fu received the B. S. degree in computer science and M. S. degree in electromechanical engineering from the University of Chongqing, Chongqing, China, in 2001 and 2006, respectively. He is currently a Doctoral postgraduate of the University of Chongqing from March 2007. His theoretical research interests include networked control systems, nonlinear control, and predictive control.

The Numerical Approximation of the Linear Advection Equation in One Space Dimension

Hongxia Li
Zhejiang University of Finance and Economics, Hangzhou, Zhejiang, 310018, China
Email: hxli@zufe.edu.cn

***Abstract*— In this paper, we will consider the linear advection equation in one space dimension. Numerical analysis the efficiency of the minmod scheme, MC scheme, Valeer scheme the super bee scheme, Beam warming scheme, Fromm scheme. Some numerical experiments are presented to show the efficiency of the schemes.**

***Index Terms*—numerical analysis, advection equation, the efficiency**

I. INTRODUCTION

In this paper, we will consider the linear advection equation in one space dimension

$$u_t + u_x = 0, u(x,0) = u^0(x) \tag{1}$$

Equation (1) is a scalar, linear, constant-coefficient PDE of hyperbolic type. The general solution of this equation is very easy to determine. Any smooth function of the form

$$u(x,t) = u^0(x-t) \tag{2}$$

are the exact solutions of (1).

There are so many numerical schemes to simulate (1), [2], [3], [4], then we will focus on higher resolution schemes: minmod scheme, MC scheme, superbee scheme and Vanleer scheme, Beam warming scheme, Fromm scheme.

II. SEVERAL NUMERICAL SCHEMES

Four high resolution schemes of Godunov's type for linear advection equation are presented in this section.

A. The steps to construct the Godunov's schemes:

Reconstruction:

We use the numerical solution $\{u_j^n\}$ to reconstruct a piecewise linear function $R(x;u^n)$ in each grid cell $[x_{j-1/2}, x_{j+1/2}]$,

$$R(x;u^n) = u_j^n + \theta_j^n (x - x_j) \tag{3}$$

where θ_j^n is the slope. It is well known that the computation of the slope in Godunov-type schemes must subject to certain stability requirements. These may be keeping the total variation of the solution diminishing(TV D).

Evolution:

Evolve the hyperbolic equation with the reconstructed solution $R(x;u^n)$ as the initial value at time t_n,

$$\begin{cases} v_t + f(v)_x = 0, -\infty < x < +\infty, t_n < t < t_{n+1} \\ v(x,t_n) = R(x;u^n), -\infty < x < +\infty \end{cases} \tag{4}$$

to obtain the solution $v(x,t)$ over the time interval $[t_n, t_{n+1}]$.

Averaging:

After obtaining $v(x,t)$ over $[t_n, t_{n+1}]$, we average it over grid cells at time t_{n+1} to obtain

$$u_j^{n+1} = \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} v(x, t_{n+1}) dx \tag{5}$$

In practice we use the integral form of (4) over the cell instead to compute u_j^{n+1}, which results in the scheme:

$$u_j^{n+1} = u_j^n - \lambda(\hat{f}_{j+1/2}^n - \hat{f}_{j-1/2}^n) \tag{6}$$

with $\hat{f}_{j\pm1/2}^n$, the numerical fluxes, being

$$\hat{f}_{j\pm1/2}^n = \frac{1}{\tau} \int_{t_n}^{t_{n+1}} f(v(x_{j\pm1/2}, t)) dt \tag{7}$$

B. Higher resolution schemes

For the linear advection equation the flux-limit method takes the form:

$$u_j^{n+1} = u_j^n - \lambda(u_j^n - u_{j-1}^n) - \frac{1}{2}\lambda(1-\lambda)[\phi(\theta_{j+1/2}^n)(u_{j+1}^n - u_j^n) - \phi(\theta_{j-1/2}^n)(u_j^n - u_{j-1}^n)] \tag{8}$$

where $\theta_{j\pm1/2}^n = \dfrac{u_{j-1}^n - u_{j-2}^n}{u_j^n - u_{j-1}^n}$. High-resolution limiters:

$$\begin{cases} \text{minmod: } \phi(\theta) = \min \text{mod}(1, \theta) \\ \text{superbee: } \phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta)) \\ \text{MC: } \phi(\theta) = \max(0, \min((1+\theta)/2, 2, 2\theta)) \\ \text{Van Leer: } \phi(\theta) = \dfrac{\theta + |\theta|}{1 + |\theta|} \\ \text{Beam Warming: } \phi(\theta) = \theta \\ \text{Fromm: } \phi(\theta) = \dfrac{\theta + 1}{2} \end{cases} \tag{9}$$

The VanLeer limiter listed here was proposed in [5]. Where the minmod function of two arguments is defined by

$$\min \text{mod}(a,b) = \begin{cases} a, if\ |a| < |b|, ab > 0, \\ b, if\ |b| < |a|, ab > 0, \\ 0, if\ ab \le 0 \end{cases} \tag{10}$$

doi:10.4304/jcp.7.1.272-277

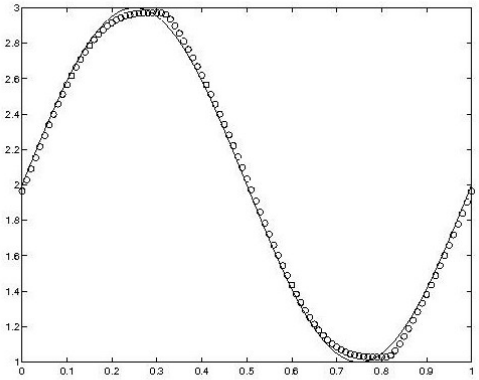
III. NUMERICAL EXAMPLES

Consider the initial problem $u_t + u_x = 0, u(x,0) = u^0(x)$ The Figs in this section: the solid line is exact solution and the dot lines represent the numerical solution.

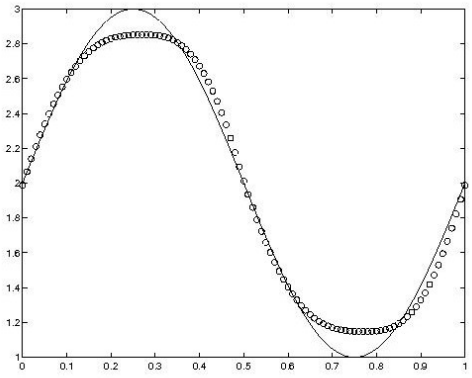
Example1:

$$u(x,0) = \sin 2\pi x, 0 \le x \le 1$$

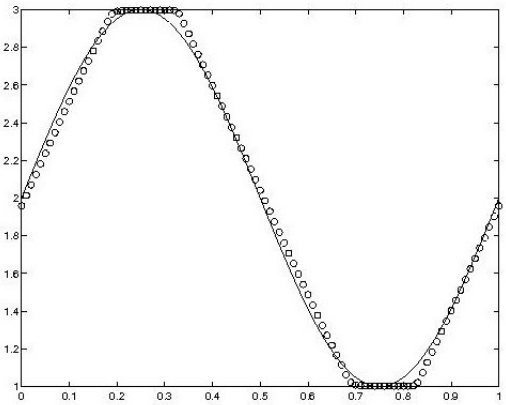
Periodic boundary condition, where h=0.01, $\lambda = 0.5$, t=100.



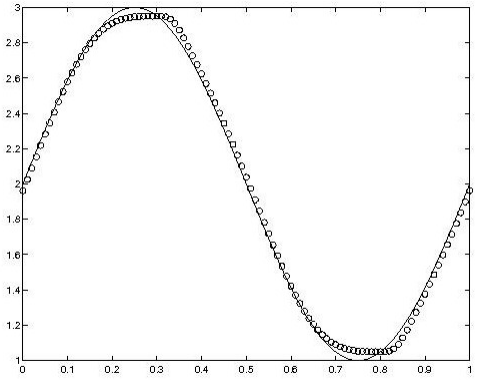
(a) MC



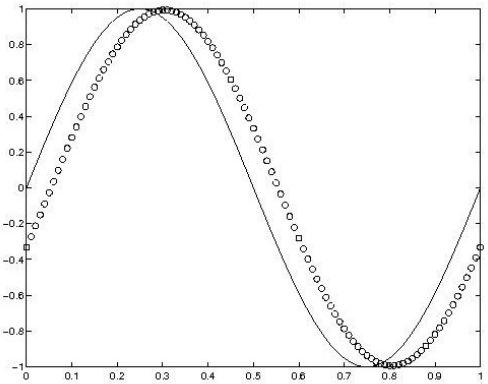
(b) minmod



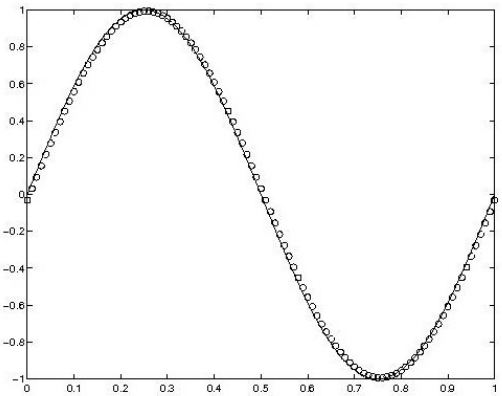
(c) superbee



(d) Van Leer



(e) Bean warming



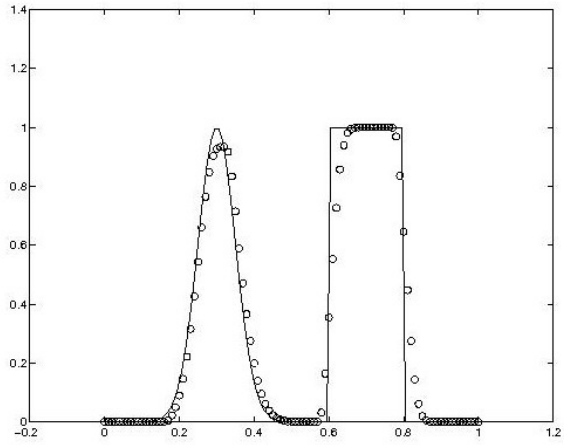
(f) Fromm

Fig1: Example1, h=0.01, $\lambda = 0.5$, t=100.

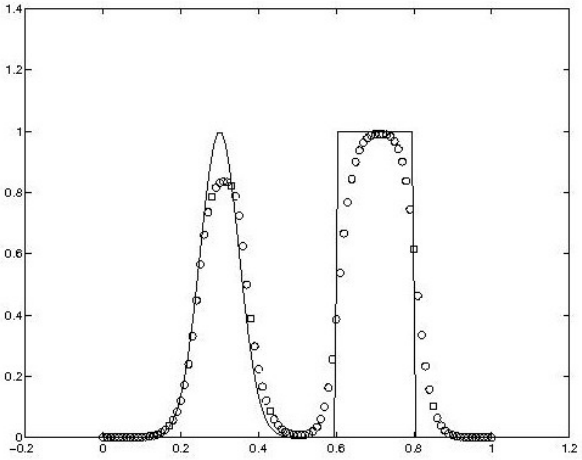
Example2:

$$u_0(x) = \begin{cases} e^{-200(x-0.3)^2}, 0.2 \le x \le 0.4 \\ 1.0, 0.6 \le x \le 0.8 \\ 0.0, otherwise \end{cases}$$

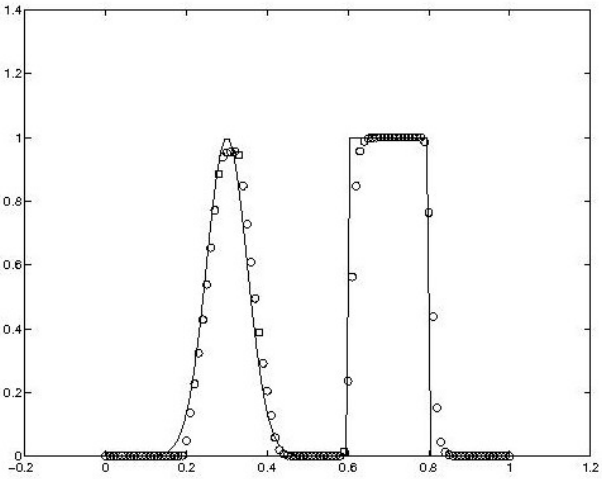
for h=0.01, t=5, $\lambda = 0.5$.



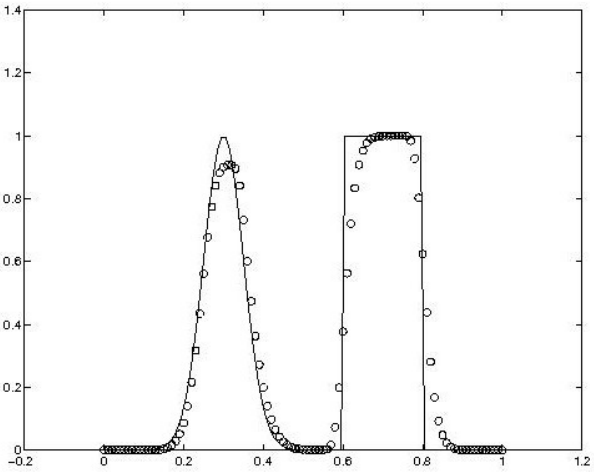
(a) MC



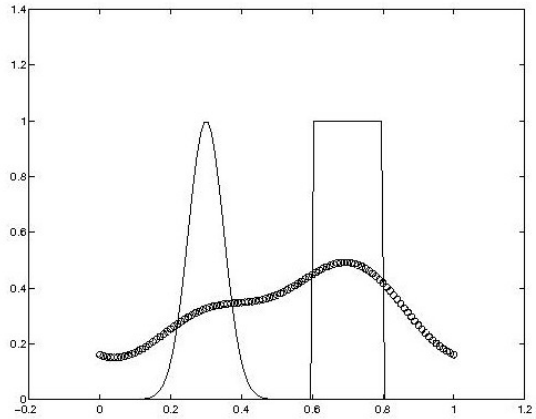
(b) minmod



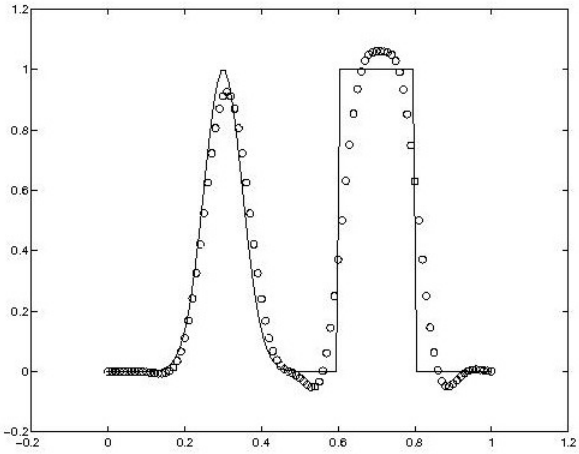
(c) superbee



(d) Van Leer



(e) beam warming



(f) Fromm

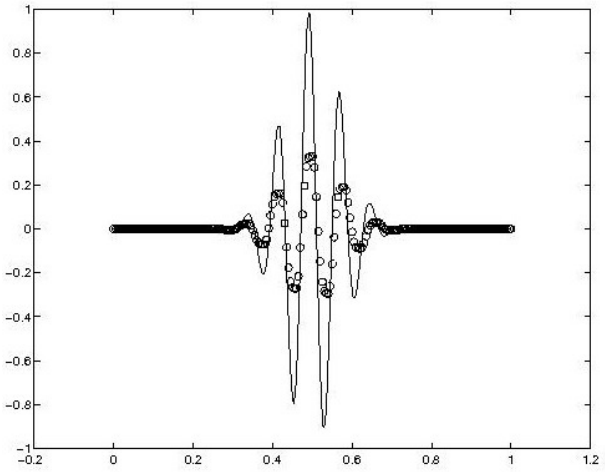
Fig2: Example2, h=0.01, $\lambda = 0.5$, t=5.

Example3:

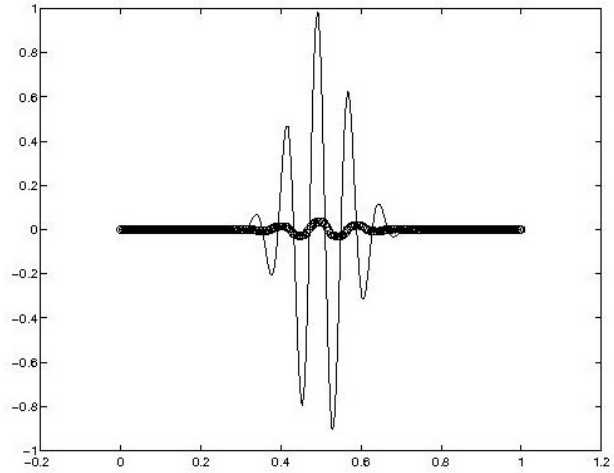
$u_0(x) = e^{-100(x-0.5)^2} \sin(80x), 0 \le x \le 1$

periodic boundary condition is implemented at the two ends. The numerical solutions on grid of 200 cells

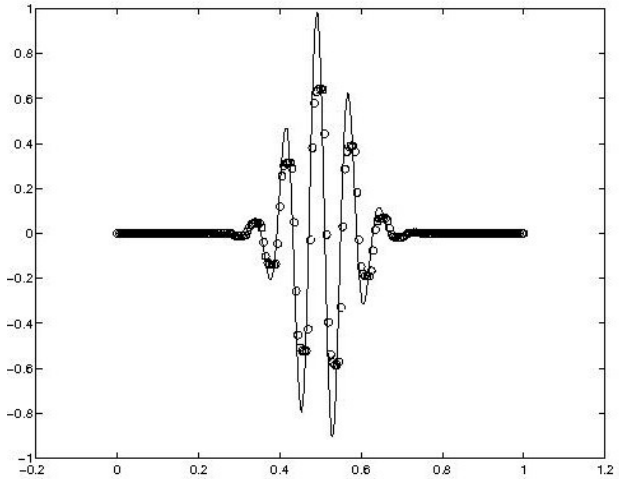
(h=0.005) with $\lambda = 0.6$ at time t=10 are displayed respectively in Fig3.



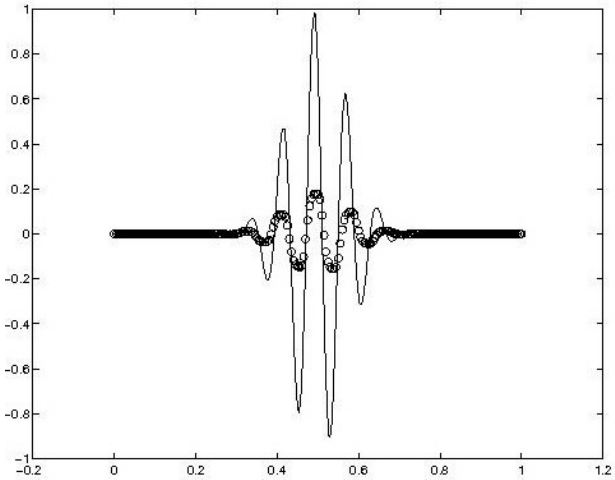
(a) MC



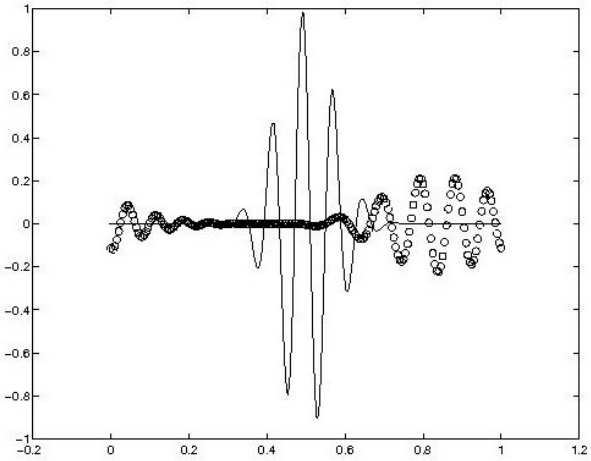
(b) minmod



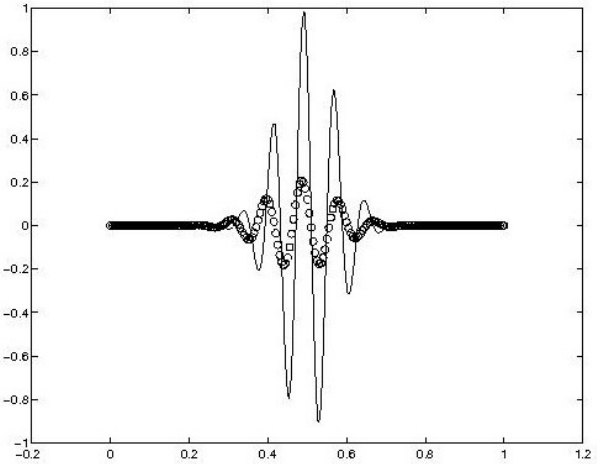
(c) superbee



(d) Van Leer



(e) Beam warming



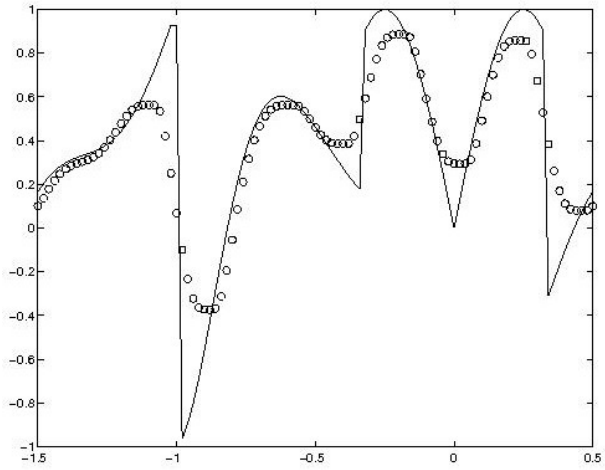
(f) Fromm

Fig3: Example3, h=0.005, $\lambda = 0.6$, t=10.

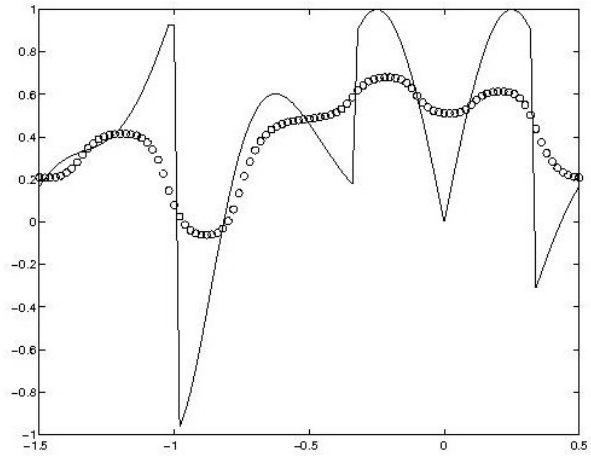
Example4:

$$u_0(x)=\begin{cases} -x\sin(\frac{3\pi x^2}{2}), -1\le x\le 1/3 \\ |\sin(2\pi x)|, |x|\le 1/3 \\ 2x-1-\frac{1}{6}\sin(3\pi x), 1/3<x\le 1 \end{cases}$$

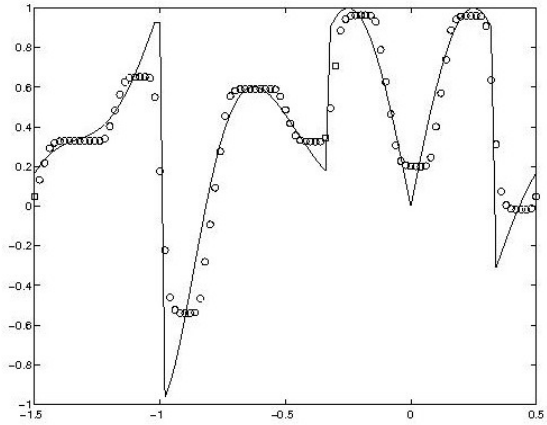
Periodic boundary condition, where h=0.02, $\lambda=0.8$, t=10. This example is first suggested by Harten in [1] and has been used by many people for their numerical tests.



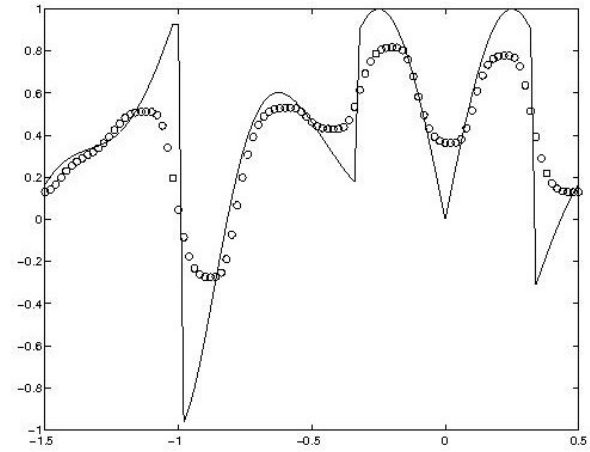
(a) MC



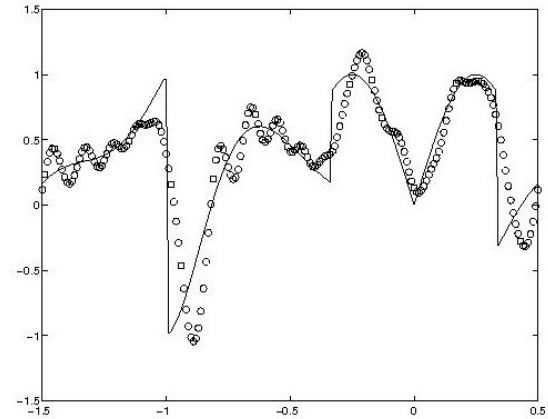
(b) minmod



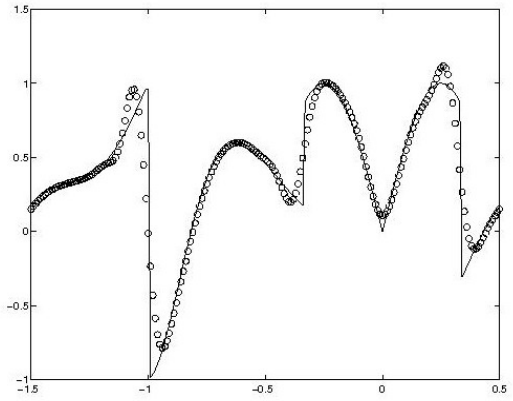
(c) superbee



(d) Van Leer



(e) Beam warming



(f) Fromm

Fig4: Example4, h=0.02, $\lambda = 0.8$, t=10.

VI. CONCLUSION

In this paper, we considered the linear advection equation in one space dimension. Numerical analysis the efficiency of the minmod scheme, MC scheme, Valeer scheme and the super bee scheme. At last some numerical experiments are presented to show the efficiency of the schemes.

ACKNOWLEDGMENT

This work was supported by China NSF Grant No. 10871168.

REFERENCES

[1] A. Harten, and S.Osher, "Uniformly high-order accurate nonoscillatory scheme I", SIAM J. Numer. Anal., 24 (1987), pp. 279-309.

[2] K.W.Morton; D.F.Mayers," Numerical solution of partial differential equations" Posts Telecom Press (2006).

[3] Randall. Leveque, "Finite Volume methods for hyperbolic problems" Cambridge university press (2002)

[4] B. Van Leer,"Towards the ultimate conservation difference schemeII: Monotonicity and conservation combined in a second order scheme", J. Comput. Phys., 14 (1974), pp. 361-370.

[5] J.W.Thomas, "Numerical partial differential equations-finite difference methods", Spring-verlag New York Berlin Heidelberg (1995).

Hongxia Li Hebei Province, China. Birthd ate: Oct, 1976. is computational mathematics Ph.D., graduated from Dept. mathematics Sha nghai University . And research interests on nu merical solutions of partial differential equatio ns. She is a associate professor of colle. Math ematics and statistics zheijing University of fin ance and economics.

The Design for Feed Water System of Boiler Based on Fuzzy Immune Smith Control

GuiLi YUAN
School of Control and Computer Engineering, NCEPU, Beijing, China
Email: ygl@ncepu.edu.cn

JiZhen LIU
School of Control and Computer Engineering, NCEPU, Beijing, China
Email: xyg@ncepu.edu.cn

***Abstract*—Aiming at the uncertainty of model parameters and dominate time delay of the controlled object in feed water system of boiler, combined with improved Smith predictor control method and fuzzy immune PID control method, improved Smith predicting controller based on fuzzy immune is designed.**

Fuzzy Immune PID controller can self-tune parameters, and has adaptive capacity to the diversification of controlled object parameters. When there is the phenomenon of large delay, the predictor estimates in advance the dynamic characteristics of the process under basic disturbance, so that regulator takes action ahead of time to reduce the overshoot. The designed controller is applied to the feed water cascade control system, and is simulated by Matlab under different operating conditions. Simulation results show that the designed control system has strong adaptive ability to the diversification of model parameters. And its stability, accuracy are superior to that of conventional Smith control system. The effectiveness of this designed controller has been confirmed.

***Index Terms*—immune control, feed water system, smith control, fuzzy immune control**

I. INTRODUCTION

The maintenance of the water level of boiler drum in a certain range is a major indicator which maintains the safe operation of turbine and boiler. Too low or too high water level will affect the operation safety, economy of turbine and boiler. At present, water level control strategy has a variety of sophisticated control methods including single-impulse control method, two impulses control method and three impulses control method, and so on. However, in a specific industrial field, the high automatic input rate for water level control can not be guaranteed. So a strong robustness, relatively simple and fast algorithm is in urgent need [1]. Smith predicting compensation method is widely used in the time delay system, but too sensitive for model diversification. In this study, the improved Smith predicting compensation control strategy based on fuzzy immune will be applied to water level control of drum. This strategy gives full play to robustness and rapidity of fuzzy control and immune control, and improves the sensitivity to the model. Simulation results show that the boiler drum water level control system using this new strategy has good dynamic adjustment quality and strong robustness.

Project number: 2007AA04Z163

II. IMMUNE CONTROL

A. Immune Feedback Mechanism

Immunization is a characteristic physiological response of the organism. Biological immune system can produce antibodies against invading antigens. After antibody and antigen binding, a series of reactions will occur. Antigen was destroyed by phagocytes or special enzymes effect. Immunization includes humoral immunity and cellular immunity. Immune response is completed by the interaction between different sub-groups T cell. T_H cell (helper T - cell) and T_S cell(suppressor T - cell) are important immune regulating cells. T cell plays a key role in the immune response ,which mainly draw on feedback regulation mechanism of T cell. For simplicity, we mainly consider the reaction between B cell and T cell, that is, the response between antigen (A_g), antibody (A_b), B cell, supporting T cell (T_H) and suppressor T cell (T_s). Taking humoral immune response as an example, the APC (Antigen Presenting Cell) digests antigen. First of all, it activates T_H cell, and releases lymphocyte. Then the B cell is activated to produce antibody. Antigen-presenting of APC can slowly activate T_s cell, which can inhibit T_H cell and B cell to ensure the stability of the immune system. This is the principal feedback mechanism above. The interaction between inhibition mechanism and the main feedback mechanism applies the rapid response of the immune

doi:10.4304/jcp.7.1.278-283

feedback mechanism to the antigen, and the rapid stability of the immune system to achieve. The regulation of T cell in immune response is shown in Fig. 1 [2].

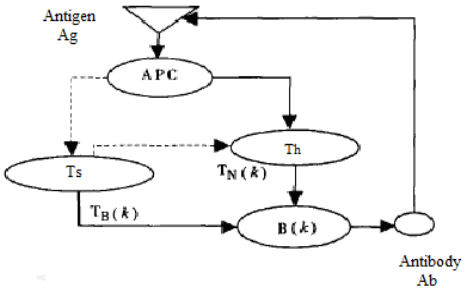


Figure. 1 Response chart of B cells in humoral immunity

B. *Immune Feedback Control Algorithm*[2,3,4,5]

Based on the immune feedback principle, we consider the following simple feedback mechanism. Supposing the number of the k generation antigens: $\varepsilon(k)$, the output of antigen-stimulated T_H cell: $T_H(k)$, and the Impact T_S cell to B cell is $T_s(k)$. Then the total stimulus $S(k)$ that B cell receives:

$$S(k) = T_H(k) - T_S(k) \tag{1}$$

Where:

$$T_H(k) = k_1\varepsilon(k), T_s(k) = k_2 f(\Delta S(k))$$

If the number of antigens $\varepsilon(k)$ is act as the deviation $e(k)$, and the total stimulus $S(k)$ that B cell receives is act as control input $u(k)$. Then the following feedback control law:

$$u(k) = K\{1 - \eta f[\Delta u(k)]\}e(k) = k_p e(k) \tag{2}$$

Where:

$K = k_1$, K : to control reaction speed

$\eta = \dfrac{k_2}{k_1}$, η : to control stabilizing effect

$f(\cdot)$:non-linear function.

Seen from Eq. (2), the controller based on immune feedback principle is actually a non-linear P controller [1,6].

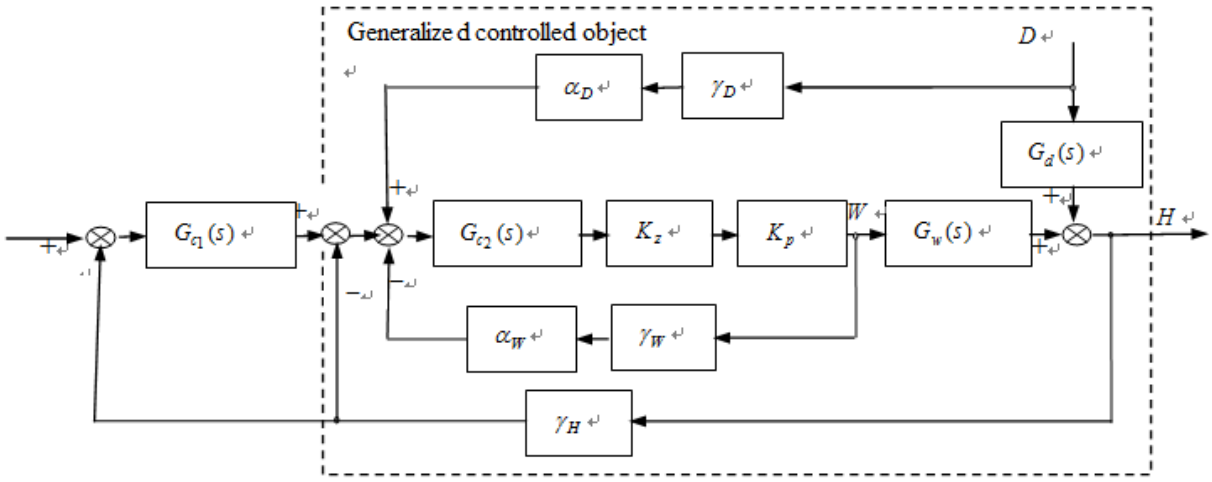


Figure. 2 The cascade three impulses control chart for drum water level

C. *Improved Smith Predicting Compensation Control*

If Smith Predicting control system can accurately predict model, this method can obtain good control effect, and greatly improve control quality. But the thermal control system with a large delay is difficult to establish accurate mathematical model. The Smith Predicting Controller is very sensitive to model error, so it is difficult to widely use in the thermal control. Improved Smith predictor proposed by Hang, it adds in a regulator based on the original program. The block diagram are shown in Fig. 2, Kp＝1[7].

Seen From the Fig. 2, the main difference between improved Smith predictor and Smith predicting compensation strategy is transfer function of feedback channel in the main feedback loop is not 1 but $G_f(s)$.

$$G_f(s) = \frac{G_{c2}(s)g_m(s)}{1 + G_{c2}(s)g_m(s)} \tag{3}$$

$G_{c1}(s), G_{c2}(s)$ are all PI regulators to ensure output response of the system has not steady-state error. The primary regulator $G_{c1}(s)$ adjusts on the basis of the circumstances that models exactly match. Auxiliary regulator setting seems to be complex, but auxiliary regulator in the feedback channel constitutes $G_f(s)$ with model transfer function $g_m(s)$. Assuming that $g_m(s)$ is a first-order inertia link $\dfrac{1}{T_m s + 1}$, and integral time constant in $G_{c2}(s)$ is $T_{i2} = T_m$. That is,

regulator integral time is equal to time constant of the model. $G_f(s)$ can be simplified as:

$$G_f(s)=\frac{1}{\frac{T_m}{K_{c2}K_m}+1}=\frac{1}{T_f s+1} \tag{4}$$

Feedback loop has a first-order filter which has only one tuning parameter T_f. Seen from Eq. (4), only the proportional gain K_{c2} in $G_{c2}(s)$ need to be set, and it is easier to adjust on line. This paper simulates by using such a control strategy in the feed water control process, in order to further understand the improved strategy impact on system performance.

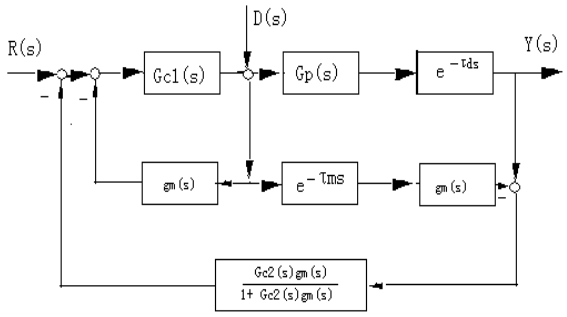


Figure. 3 The block diagram of improved Smith compensator system

III. Feed Water of Boiler Control System

Water level cascade three impulse control system are shown in Fig. 3.

In the Fig. 3:

H, H_0: water level output value and the given value

D, W: steam flow and feed water flow

$G_w(s), G_d(s)$: the transfer function of water flow to water level and the transfer function of steam flow to water level:

$$G_w(s)=\frac{\varepsilon}{s(Ts_1+1)}, G_d(s)=\frac{K_d}{T_2 s+1}-\frac{\varepsilon}{s} \tag{5}$$

$\gamma_D, \gamma_W, \gamma_H$: transfer coefficient of steam flow D, transfer coefficient of feed water flow W and the transfer coefficient of water level measurement transmitter H

α_D, α_W: sub-pressure coefficient of steam flow and sub-pressure coefficient of feed water flow K_Z, K_P: characteristic coefficient of implement agencies and the characteristic coefficient of valve [6].

Comparing to the general single-stage three impulse water level control system, two regulators complete water control task in this control system. Vice regulator $G_{c2}(s)$ task is to eliminate spontaneous disturbance of water flow caused by pressure of feed water fluctuation, and quickly adjusts feed water flow when the steam load changes, to ensure the balance between feed water flow and steam flow. The primary regulator $G_{c1}(s)$ task is to ensure the water level without the static error. Feed-forward channel is used to compensate for external disturbances to overcome the "false water level" phenomenon. Thus, when the load changes, the water level stable value depends on the primary regulator to maintain, and it does not require the steam flow signal intensity which enter to the vice regulator carry out the set according to "static ratio" [6].

The parameters of object in Fig. 3:

$\varepsilon = 0.037, K_d = 3.6, T_1 = 30, T_2 = 15, \gamma_D = \gamma_w = 0.08,$

$\gamma_H = 0.033,$

$\alpha_D = \alpha_w = 0.21, K_z = 10, K_P = 2$

When the controller $G_{c2}(s)$ takes $K_{c2} = 0.4648$, the approximated, generalized controlled object model and disturbance output model respectively are [1]:

$$G_{mv}(s)=\frac{30.303}{80s+1}e^{-25s}, G_{md}(s)=\frac{-3.25}{60s+1}e^{-100s} \tag{6}$$

IV. The Design of Improved Smith Controller Based on Fuzzy Immune

Taking into account the difficulty of the selection of parameters K, η and non-linear function $f(\cdot)$, this paper applies fuzzy logic reasoning to approximate the nonlinear function $f(\cdot)$, then adjusts real-time the proportional gain K_p of PID controller based on fuzzy immune feedback mechanism. The composition form of primary controller PID is: $G_c(s) = k_p + k_i \frac{1}{s} + k_d s$.

In the dynamic process, proportional, integral and differential play different roles. The relationship between the performance in time domain and k_p, k_i, k_d is shown in Table.1.

Table Ⅰ.
The Relationship between Adjusting Parameters of PID and the Performance in Time Domain

Parameter	Rise time	Overshoot	Transition time	Static error
k_p increase	decrease	increase	decrease of variation	decrease
k_i increase	decrease	increase	increase	elimination
k_d increase	decrease of variation	decrease	decrease	decrease of variation

Seen from Table.1, when PID parameters increase, the performance of system makes the changes. The relationship in Table. 1 is relative in a certain range. k_p, k_i, k_d, the change for every parameter takes different effect on system performance.

At the same time, this paper applies fuzzy controller to tune online integral coefficient k_i and derivative coefficient k_d of PID controller, to achieve adaptive control [2,7].This paper introduces the framework of the control system, which is a new control strategy combined

by the controller and improved Smith predictor. The framework of the control system is presented in Fig. 4.

Using fuzzy rules to approximate the nonlinear function f (•), each input variable is fuzzed by two fuzzy sets, respectively the "positive" (P), "negative" (N). Output variable is fuzzed by three fuzzy sets, namely, the "positive" (P), "Zero" (Z) and "negative" (N).

Membership function defined on (- ∞, + ∞). According to "the greater stimulus that cell accepts, the smaller the suppression ability is" and" the smaller stimulus that cell accepts, the greater the suppression ability is ", the following four fuzzy rules are used [4].

r1: If u is NB and Δu is NB, then f (u, Δu) is PB.

r2: If u is NB and Δu is NM, then f (u, Δu) is PB.

r3: If u is NB and Δu is NS, then f (u, Δu) is PM.

r4: If u is NB and Δu is ZO, then f (u, Δu) is PM.

The membership functions of u, Δu, and f (•) are shown in Fig. 5.

In the Fig. 5:

PB: the positive big.

NB: the negative big.

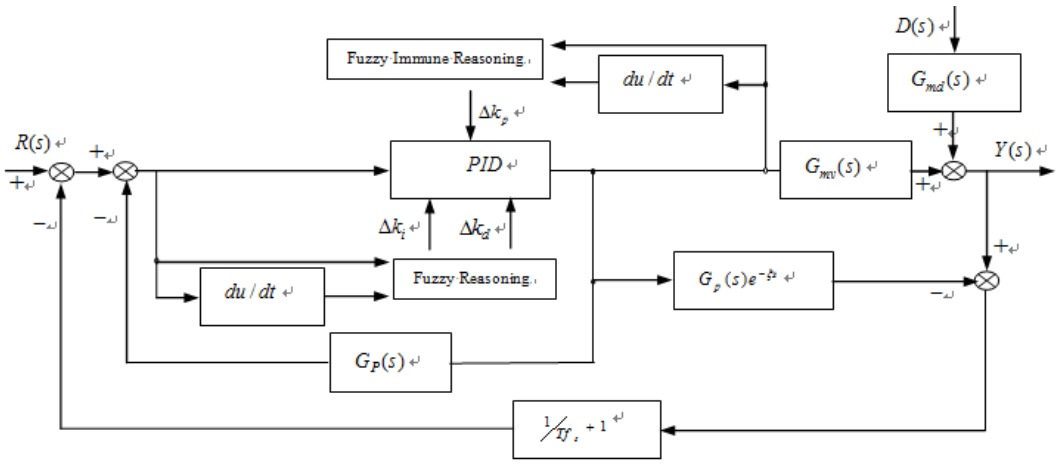


Figure. 4 The improved Smith control chart based on fuzzy immune for boiler drum water level

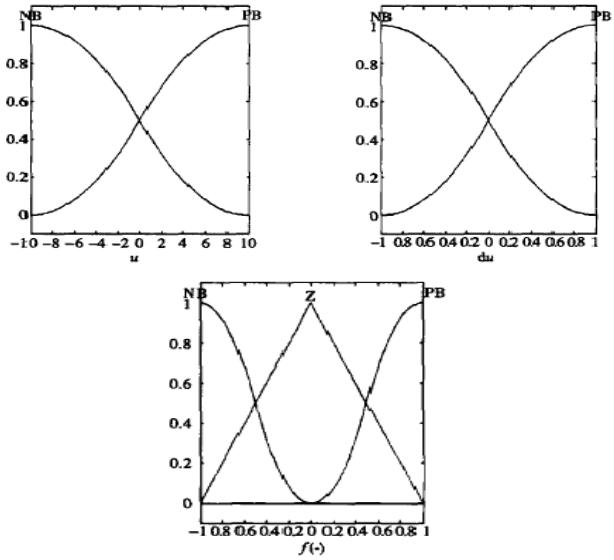


Figure. 5 The membership function for u, du, and f (.) of the fuzzy immune PID

When fuzzy control rule table for k_i, k_d has been built as shown in Table. 2 and Table. 3, according to the following method, we can do adaptive correction on k_p, k_i, k_d. The range of the system error e and the range for change rate of error ec are defined as field on fuzzy sets, that is, e,ec={-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6}. Fuzzy subset e,ec＝{NB,NM,NS,ZO,PS,PM,PB}, elements of subset respectively represent the negative large, negative, negative small, zero, positive small, positive middle, positive large. Supposing e, ec, and k_p, k_i, k_d are subject to normal distribution, then the membership of each subset can be obtained. According to membership assignment form of the fuzzy sets and the fuzzy control model, we design fuzzy matrix for PID parameters using fuzzy synthetic reasoning, and find modified parameters which are used into the following formula.

$$\begin{aligned} k_p &= k_p' + \{e_i, ec_i\}_p \\ k_i &= k_i' + \{e_i, ec_i\}_i \\ k_d &= k_p' + \{e_i, ec_i\}_d \end{aligned} \tag{7}$$

TABLE Ⅱ.

FUZZY RULE OF k_i

e / ec	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

TABLE III.
FUZZY RULE OF k_d

e \ ec	NB	NM	NS	ZO	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	NS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

In the online operation process, the control system complete the online self-correction on PID parameters of the main controller, through the results processing of the fuzzy logic rules, table and computing.

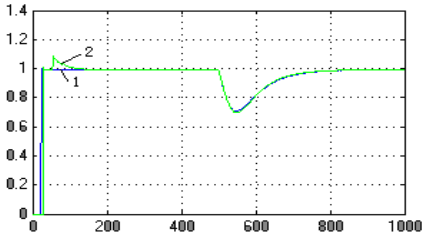
According to the control scheme proposed in this paper, we take the object model of generalized feed water in Fig. 2 and the disturbance model into the control system structure diagram in Fig. 4 using Eq. (6). This paper uses the advantage that fuzzy control can approximate an arbitrary non-linear link and robustness, to apply control experience of the operator and expert to the control process, and draw on speed of immune feedback response and immune stability to improve the control quality. It makes control scheme have good rapidity, robustness and adaptive capacity.

V. SIMULATION

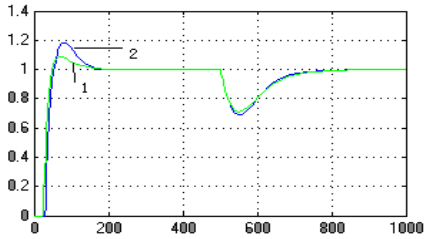
This paper applies Simulink tools to program, then to do analysis, calculation and simulation work easily and intuitively.

When the object model matches, fuzzy immune Smith control scheme and Smith control scheme are used to simulate respectively. Then simulation results are shown from Fig. 6 to Fig. 9. The figures respectively corresponds to two schemes when delay time constant increased by 10%, the delay time constant increased by20%, the inertia time constant increased by 10%, the inertia time constant increased by 20%. Reaching the steady state, at t = 500s, the 20%disturbance of steam flow has been added. From Fig. 6 to Fig. 9, chart (a) and chart (b) respectively represent response curves comparison using fuzzy immune Smith scheme and Smith control scheme. Curve 1 and curve 2 respectively represent the comparison between the response curve for the no-changed parameters and the response curve for the changed parameters using the corresponding control strategy. Fig. 6(a) is the response curve using fuzzy immune Smith when delay time increase by 10%. Curve 1 is the response curve for nominal parameters, but curve 2 represents the response curve when the delay time increases by 10%. So, seen from the comparisons, when the model match or mismatch, the fuzzy immune Smith control has small overshoot and the short regulating time. And, when the system occur the disturbance of the set value or the external load disturbance, Smith predictor control is very sensitive to the accuracy of the model, but fuzzy immune Smith program has excellent adaptability and robustness.

If the expert experience of fuzzy control rules can be detailed to make more effective control rules, the control effect of Smith predicting controller based on fuzzy immune will be excellent.

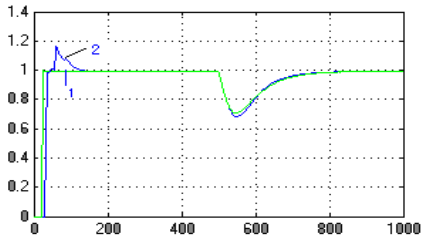


6 (a)

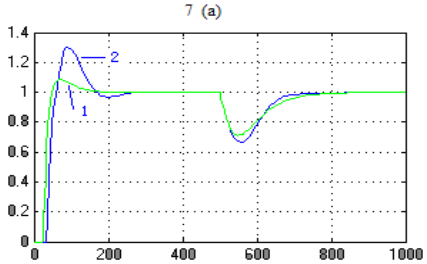


6 (b)

Figure. 6 Simulation result that τ increases by 10%

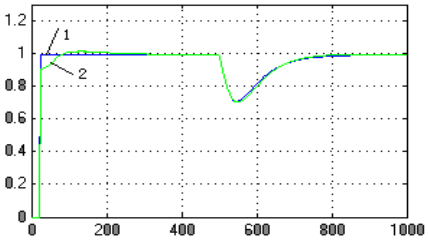


7 (a)

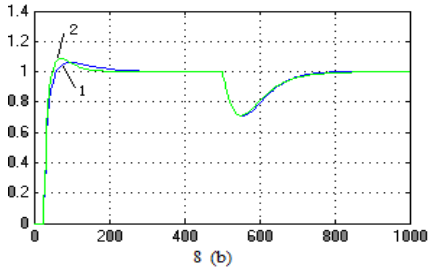


7 (b)

Figure. 7 Simulation result that τ increases by 20%

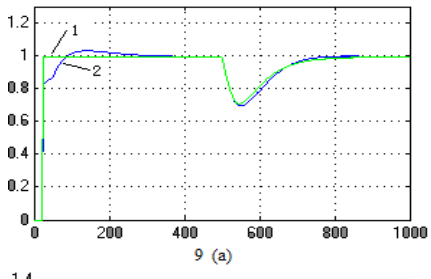


8 (a)

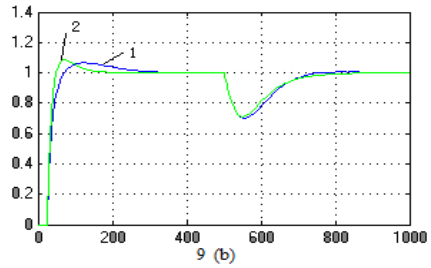


8 (b)

Figure. 8 Simulation result that T increases by 10%



9 (a)



9 (b)

Figure. 9 Simulation result that T increases by 20%

VI. CONCLUSION

Seen from Fig. 6, Fig. 7, Fig. 8 and Fig. 9, Smith control scheme, improved Smith predicting control program and improved Smith control scheme based on fuzzy immune, regardless of the changes appear in the model match or the model parameters, the control performance is improved step by step. Especially improved Smith control scheme based on fuzzy immune presented in this paper can significantly improves adjustment time, the overshoot, and robust stability. It also effectively improve the system performance of control system for coordination feed water.

If the expert experience of fuzzy control rules can be detailed to make more effective control rules, the control effect of Smith predicting controller based on fuzzy immune will be excellent.

ACKNOWLEDGMENT

This work was supported in part by Ministry of Science and Technology 863 Project (2007AA04Z163).

REFERENCES

[1] D.F. Wang, P. Han and G.Y. Wang, "Predictive functional control for water level of boiler drum," *Journal of North China Electric Power University*, vol. 30(3), pp. 44 -47, 2003.

[2] N. Wang, G. Ren, "Study of ship boiler fuzzy immune self-adapting PID controller with system simulation,"*Journal of Dalian Maritime University*, vol.32(1), pp.1-4, 2006.

[3] X.G. Fu, "Application of the Fuzzy Immune PID Control in Industrial Boiler Control System,"*Boiler Technology*,vol.37(4), pp.24-27, 2006.

[4] Z.Q. Ge, B.C. Qu, Q. Li, "Main Steam Temperaturs Control System Based on Fuzzy Immune Smith PID Cotroller," *Computer Simulation*, vol.23(6), pp.233-235, 2006.

[5] Y.X. Yang; D. Liu; Y. Gao, "Main Steam Temperaturs Control System in Fossil Fired Power Station Based on Fuzzy Immune PID-Smith Controller," *Acta Simulata Systematica Sinica*, vol.17(11), pp.2756-2758, 2005.

[6] D.G Peng, P. Yang, "Single neuron self-adapting compensation control via Smith predictor for drum level in power station," *East China Power*, vol. 32(12), pp.8-11, 2004.

[7] L.X. Bian, R.H. Zhao, "Main Steam Temper ature Control System Based on Fuzzy Self-tuning PID Smith Controller," *Computer Simulation*, vol. 21(1), pp.102-104, 2004.

GuiLi Yuan NeiMengGu Province, China. Birthdate: October, 1971. is Control Science and Engineering Ph.D., graduated from School of Control and Computer Engineering North China Electric Power University. And research interests on information control, advanced control strategy and its application in power plant control.

She is a associate professor of School of Control and Computer Engineering North China Electric Power University.

JiZhen Liu Shanxi Province, China. Birthdate: Aug, 1951. is Control Science and Engineering M.E, graduated from School of Control and Computer Engineering North China Electric Power University. And research interests on optimal control of power plant.

He is a professor, and PhD supervisor of School of Control and Computer Engineering North China Electric Power University.

A Grey Model for Evaluation of Information Systems Security

Huawang Shi
School of Civil Engineering, Hebei University of Engineering, Handan, P.R.China
Email: stone21st@163.com

Yong Deng
School of Electronics and Information Technology, Shanghai Jiao Tong University, Shanghai, China
Email: stone21st@163.com

***Abstract*—Quantitative security risk evaluation of information systems is increasingly drawing more and more attention. The purpose of this paper is to propose a novel method integrated grey relational analysis and grey-AHP evaluation to classification for information systems (IS) security. There are, of course, multiplicities of factors that will affect the security evaluation of information systems. Using grey relational analysis, we provided a tool to aid clients and their consultants in estimating or benchmarking the information systems security. It then provides a grey evaluation model of estimating the indicator system of information systems on the basis of the related reference, in which an evaluation methodology based on combination of grey evaluation method and Group-decision AHP method(Grey-AHP) for classifying grey clusters, calculating weights, creating an evaluation matrix and using comprehensive coefficient are presented. An example of practical application is given to show the effectiveness of this method. The result is believed to provide new means and ideas for the evaluation of IS security.**

***Index Terms*—information systems, security evaluation, grey relational analysis, grey-AHP evaluation**

I. INTRODUCTION

The rapid and dramatic advances in information technology (IT) in recent years have without question generated tremendous benefits. At the same time, however, they have created significant, unprecedented risks to government operations. Computer security has, in turn, become much more important as all levels of government utilize information systems security measures to avoid data tampering, fraud, disruptions in critical operations, and inappropriate disclosure of sensitive information. Such use of computer security is essential in minimizing the risk of malicious attacks from individuals and groups. To be effective in ensuring accountability, auditors must be able to evaluate information systems security and offer recommendations for reducing security risks to an acceptable level. To do so, they must possess the appropriate resources and skills.

Your organization may be vulnerable to attack from the outside or the inside if you remain unaware of security issues, simply ignore them or don't sufficiently manage them. An attack may take down your network or lead to the theft of sensitive data — customer information, employee information or intellectual property. The ensuing loss of public trust or the failure to comply with regulations could result in severe financial repercussions. A major security breach could also cause irrevocable damage to your organization's reputation. The benefits of information security assessment are listed as follow[1]:

(1) Provides a clear understanding of current information security risks. (2)Identifies the potential impact of vulnerabilities on your network infrastructure. (3) Raises internal awareness of information security risks. (4)Enables more informed decision-making and identifies the gaps in organizational security controls, policies and processes. (5) Provides a specific, actionable plan to improve overall security posture based on business needs. (6) Enables you to proactively address security issues before they are exploited. (7) Helps to meet regulatory compliance requirements.

The evaluation of information systems (IS) security is a process in which the evidence for assurance is identified, gathered, and analyzed against criteria for security functionality and assurance level. This can result in a measure of trust that indicates how well the system meets particular security target. However, as the information systems complexity increases, it becomes increasingly hard to address security targets and the concept of perfect security proves to be unachievable goal for computer systems developer, testers and users [2][3].

Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization. Risk assessment is the first process in the risk management methodology. Organizations use risk assessment to determine the extent of the potential threat and the risk associated with an information system. The output of this process helps to identify appropriate controls for reducing or eliminating risk during the risk mitigation process. Early evaluation of IS is actually assessment of the decision to procure IS. Decision may be rational or political, or may include features of both types. Rational decisions use systematic and accurate data, decisions are logically based on the value

doi:10.4304/jcp.7.1.284-291

maximization upon given constraints [4]. Decisions become uncertain when there are no enough data to validate them. Such a situation is quite possible at early stage. Combinations of both described decisions types are most widely used [4][5].

To evaluate the information systems security quantitatively, scholar proposed a series of theoretical analysis models. Among these models there are several influential ones, for example, AHP, grey evaluation, fuzzy comprehensive judgment, AHP-fuzzy comprehensive judgment, etc. We should review all the assessment methods in order to find the most suitable method for assessing the security of information systems. Some experts have studied these. We carried out a comprehensive evaluation of a information systems using a grey analysis. First, we confirmed the weight of each index quantitatively by mean s of Group-decision AHP according to an established index system. Then, we defined the elements of an assessment matrix using fuzzy and a quality assessment model for construction project is set up. The advantage of this approach is that it does not rely on the experience of experts an d it can improve the validity and the precision of evaluation. Consequently, it can reflect the quality status of teaching quality in higher institutions.

The purpose of this paper is to introduce and illustrate a new information systems security assessment model called the grey comprehensive evaluation and attribute hierarchy model (AHM). In Section Ⅱ of the paper we introduced the grey system theory including grey relational analysis and Grey-AHP evaluation model . In Section Ⅲ of the paper we describe the information systems security evaluation problem and gives the index of information systems security. We apply the grey model to the assessment of information systems security and give an application case. In Section Ⅴ of the paper we conclude content of this paper and outline future directions for research.

II. METHODOLOGY

A. Introduction to Grey System

The Grey System Theory that was developed by Deng [10] is mainly utilized to study system models uncertainty, analyze relations between systems, established models, and forecast and makes decisions [8]. Grey Relational Analysis is utilized to probe the extent of connections between two digits by applying the methodology of departing and scattering measurement to the actual measurement of distance and Grey Relational Analysis is an effective means of analyzing the relationship between two series. This study applies grey relational analysis to measure the similarity between the series.

Nowadays, the black is represented, as lack of information, but the white is full of information. Thus, the information that is either incomplete or undetermined is called Grey.

A system having incomplete information is called Grey system. The Grey number in Grey system represents a number with less complete information. The Grey element represents an element with incomplete information. The Grey relation is the relation with incomplete information. Those three terms are the typical symbols and features for Grey system and Grey phenomenon. There are several aspects for the theory of Grey system [9]:

(1)Grey generation: This is data processing to supplement information. It is aimed to process those complicate and tedious data to gain a clear rule, which is the whitening of a sequence of numbers.

(2)Grey modeling: This is done by step 1 to establish a set of Grey variation equations and Grey differential equations, which is the whitening of the model.

(3)Grey prediction: By using the Grey model to conduct a qualitative prediction, this is called the whitening of development.

(4) Grey decision: A decision is made under imperfect countermeasure and unclear situation, which is called the whitening of status.

(5)Grey relational analysis: Quantify all influences of various factors and their relation, which is called the whitening of factor relation.

(6)Grey control: Work on the data of system behavior and look for any rules of behavior development to predict future's behavior, the prediction value can be fed back into the system in order to control the system.

B. Grey Relational Analysis

Constitute the grey relational sets: $X=\{x_1, x_2, \ldots, x_m\}$

The main procedure of GRA is firstly translating the performance of all alternatives into a comparability sequence. This step is called grey relational generating. According to these sequences, a reference sequence is defined. Then, the grey relational coefficient between all comparability sequences and the reference sequence is calculated. Finally, based on these grey relational coefficients, the grey relational entropy between the reference sequence and every comparability sequences is calculated.

We build the grey relational ordinal by the following equations. Determined the number of columns to reference and the number of columns to compare Define x_i for the time needed for constructing and operating a correct and perfect product information model by the ith user.

$x_{i0}=\{x_{i0}(1), x_{i0}(2), \ldots, x_{i0}(l)\}$, $x_{i0}, x_{ij} \in X_i$

$x_{ij}=\{x_{ij}(1), x_{ij}(2), \ldots, x_{ij}(l)\}$

$\Delta_{i0j}(k)=\left| x_{i0}(k)-x_{ij}(k) \right|$

$r(x_{i0}(k),$

The sets of data must be standardized since the scales of criteria indices differ. Grey relational generating is used to replenish the message from the data by determining regularities and properties in jumbled data. Restated, grey relational generating can reduce the randomization and increase the regularity of data[11-13].

Data pre-processing is normally required since the range and unit in one data series may differ from the

others. Data pre-processing is also necessary when the series scatter range is too large, or when the directions of the target in the series are different. Data pre-processing is a process of transferring the original series to a comparable series. Depending on the characteristics of data series, there are various methodologies of data pre-processing available for the grey relational analysis. We use the following equations to pre-process the two series.

When, $x_i = \{x_i(j) | j = 1,2,\cdots N\}$

$$x_i'(j) = \begin{cases} \dfrac{x_i(j) - \min\limits_i x_i(j)}{\max\limits_i x_i(j) - \min\limits_i x_i(j)}, \\ \dfrac{\max\limits_i x_i(j) - x_i(j)}{\max\limits_i x_i(j) - \min\limits_i x_i(j)}, \\ 1 - \dfrac{|x_i(j) - c_i|}{\max\limits_i |x_i(j) - c_i|}, \end{cases} \quad (1)$$

Where, $\min\limits_i x_i(j)$ and $\max\limits_i x_i(j)$ Respectively, $i = 1,2,\cdots I$ is the minimum and maximum values for the data sequence x_i in the j th item, c_i is the data series in the j th item of a standard reference value.

$$x_{ij}(k) = \frac{\omega_{ij}(k) - \omega_{ij}(\min)}{\omega_{ij}(\max) - \omega_{ij}(\min)} \quad (2)$$

Where,

$$\omega_{ij}(\min) = \min_k \omega_{ij}(k), \omega_{ij}(\max) = \max_k \omega_{ij}(k), \omega_{ij}(avg) = \frac{1}{n}\sum_{k=1}^{n} \omega_{ij}(k)$$

$\Pi \omega_{ij}(k) \in \omega_{ij}]$ $k \in K = \{1, 2, \cdots, l\}$, $i = \{1, 2, \cdots, m\}$, $j \in J = \{1, 2, \cdots, n\}$.

Then

$$x_i = \begin{bmatrix} x_{i1}(1) & x_{i1}(2) & \cdots & x_{i1}(l) \\ x_{i2}(1) & x_{i2}(2) & \cdots & x_{i2}(l) \\ \vdots & \vdots & \ddots & \vdots \\ x_{in}(1) & x_{in}(2) & \cdots & x_{in}(l) \end{bmatrix} \quad (3)$$

After data pre-processing is carried out, a grey relational coefficient can be calculated with the pre-processed series. The gray correlation coefficient between the number of columns of comparison and the reference number of columns were calculated The grey relational coefficient is defined as follows:

$$x_{ij}(k) = \frac{\min\limits_j \min\limits_k \Delta_{ioj}(k) + \zeta \max\limits_j \max\limits_k \Delta_{ioj}(k)}{\Delta_{ioj}(k) + \zeta \max\limits_j \max\limits_k \Delta_{ioj}(k)} \quad (5)$$

The gray correlation evaluation indicators are as follows:

$$r(x_{i0}, x_{ij}) = \frac{1}{l}\sum_{k=1}^{l} r(x_{i0}(k), x_{ij}(k)) \quad (6)$$

So get gray relational matrix:

$$r = \begin{bmatrix} r(x_{10}, x_{11}) & r(x_{10}, x_{12}) & \cdots & r(x_{10}, x_{1n}) \\ r(x_{20}, x_{21}) & r(x_{20}, x_{22}) & \cdots & r(x_{20}, x_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ r(x_{m0}, x_{m1}) & r(x_{m0}, x_{m2}) & \cdots & r(x_{m0}, x_{mn}) \end{bmatrix} \quad (7)$$

In gray correlation matrix, each row vector element indicates the correlation degree of target sequences associated with its reference.

In equation (7), greater correlation coefficient means that time needed for users to establish an accurate and complete product information model and finish corresponding operation will have less difference from the expectation time. But there are many Gray coefficient calculated, each factor can only reflect the match level of corresponding construction duration model and actual construction duration model. It can not indicate the construction duration as a whole.

C. Grey-AHP Evaluation Model

1) Method of Group-decision Analytical Hierarchy Process(AHP): The Analytic Hierarchy Process (AHP) was developed by Thomas L. Saaty[19,20] in the 1970s and has been extensively studied and refined since then. It is to express a complex decision-making problem as a sequential step-up hierarchy structure, compute the comparatively weightiness measurement of diversified decision-making behaviour, scheme and decision-making object under different rule and the whole rule, and then rank them according to the measurement, providing decision-making evidence for the decision-makers[20].

The steps to solve the real problems using AHP method areas follows:

Model the problem as a hierarchy containing the decision goal, the alternatives for reaching it, and the criteria for evaluating the alternatives.

Establish priorities among the elements of the hierarchy by making a series of judgments based on pairwise comparisons of the elements. For example, when comparing potential real-estate purchases, the investors might say they prefer location over price and price over timing.

Synthesize these judgments to yield a set of overall priorities for the hierarchy. This would combine the investors' judgments about location, price and timing for properties A, B, C, and D into overall priorities for each property.

Check the consistency of the judgments.

Come to a final decision based on the results of this process.

According to the above steps to achieve the ultimate weight vector of every decision-maker respectively.

$$W_k = (w_{1k}, w_{2k}, \ldots, w_{mk})^T, k = 1,2,3,\ldots t \quad (8)$$

Assuming there is t decision-makers, the number of factors to be evaluated are m, for each given a series the determining matrix of decision-makers, then by calculating the weighted average base sequencing vector the relative weight vector can be got as: $\varpi = (\varpi_1, \varpi_2, \ldots, \varpi_m)^T$.

$$\varpi_j = \frac{\theta_j}{\sum_{i=1}^{m} \theta_i}, \quad j = 1,2,\ldots,m \tag{9}$$

$$\theta_j = (w_{j1})^{\lambda_1} \cdot (w_{j2})^{\lambda_2} \ldots (w_{jt})^{\lambda_t} \tag{10}$$

$$\sum_{k=1}^{t} \lambda_k = 1, \lambda_1, \lambda_2, \ldots, \lambda_t$$

Where, is weight coefficient of decision-maker, Specially, $\lambda_1 = \lambda_2 = \ldots = \lambda_t = \frac{1}{t}$. Finally, the standard deviation was calculated as follow:

$$\sigma_j = \sqrt{\frac{1}{t-1} \sum_{k=1}^{t} (w_{jk} - \varpi_j)^2} \tag{11}$$

When $\sigma_j < 0.5$, the group-decision is acceptable, and right-vector of the indicators is feedback to each decision-makers. If a number of decision-makers accept the weight vector, then calculation end, otherwise, ask decision-makers to propose amendments to judge views, and so repeatedly, until the weight vector is satisfactory to decision-makers.

2) Determining Grade Rank Standard of Indices:

U represents the first level evaluation factors U_i composed of a collection as $U = \{U_1, U_2, \cdots, U_6\}$. $V_j (i = 1,2,\cdots 6)$ represents the factors in the second level consisting of a collection of $V_i = \{V_{i1}, V_{i2}, \cdots, V_{in}\}$ .The grade rank vector of evaluation is :

$$V = \{v_1(best),\ v_2(good),\ v_3(average),\ v_4(worse)\}$$

Evaluation index U is a qualitative index. It is transformed to the quantification index through establishing grade rank standard of evaluation indices. Considering evaluation grade is divided into 4 levels, the grade and the evaluation points of each ranks is 4, 3, 2 and 1. If the grade between two adjacent grades, The corresponding score value of 3.5, 2.5, and 1.5 points. Specific grades are given by experts based on experience.

3) Determining Weight of Indexes U_i and V_{ij} :

It is a very important step to determine the weights of each indices of teaching quality, because the elements of the quality of teaching quality take different proportion in the comprehensive evaluation. There are many ways to determine the weights, in this paper, Method of Group-decision Analytical Hierarchy Process (AHP) was adopted(discussed in part A).

4) Establishing the Appraisal Grey Cluster :

The committee of experts scoring various indicators in accordance with measured values and professional experience. Suppose the serial number of appraisal grey is $e, e = 1,2,3,4,$ namely has 4 appraisal grey cluster. The first grey cluster is "best" （e=1）, suppose grey number $\otimes_1 \in [4, \infty)$, the corresponding grey albinism weight function is f_1. The second grey cluster is "better" （ e=2 ）, suppose grey number $\otimes_2 \in [0,3,6]$, the corresponding grey albinism weight function is f_2. The first grey cluster is "average" （e=3）, suppose grey number $\otimes_3 \in [0,2,4]$, the corresponding grey albinism weight function is f_3. The first grey cluster is "worse" （ e=4 ）, suppose grey number $\otimes_4 \in [0,1,2]$, the corresponding grey albinism weight function is f_4.

$$f_1(d_{ijm}) = \begin{cases} d_{ijm}/4, d_{ijm} \in [0,4] \\ 1, d_{ijm} \in [4, \infty) \\ 0, d_{ijm} \notin [0, \infty) \end{cases} \tag{12}$$

$$f_2(d_{ijm}) = \begin{cases} d_{ijm}/3, d_{ijm} \in [0,3] \\ 6 - d_{ijm}/3, d_{ijm} \in [3,6] \\ 0, d_{ijm} \notin [0,6] \end{cases} \tag{13}$$

$$f_3(d_{ijm}) = \begin{cases} d_{ijm}/2, d_{ijm} \in [0,2] \\ 4 - d_{ijm}/2, d_{ijm} \in [2,4] \\ 0, d_{ijm} \notin [0,4] \end{cases} \tag{14}$$

$$f_4(d_{ijm}) = \begin{cases} 1, d_{ijm} \in [0,1] \\ 2 - d_{ijm}, d_{ijm} \in [1,2] \\ 0, d_{ijm} \notin [0,2] \end{cases} \tag{15}$$

5) Calculating Grey Appraisal Weight Vector and Weight Matrix :

The grey appraisal weight of vector of e grey cluster is marked x_{ije} :

$$x_{ije} = \sum_{m=1}^{p} f_e(d_{ijm}), x_{ij} = \sum_{e=1}^{6} x_{ije} \tag{16}$$

Considering 4 grey cluster ,namely e=1,2,3,4, for each grey cluster, grey appraisal weight vector of evaluation indices V_{ij} is marked as $r_{ij}: r_{ij} = (r_{ij1}, r_{ij2}, r_{ij3,} r_{ij4})$ and the evaluation weight matrix is R_i ,then:

$$R_i = \begin{bmatrix} r_{i1} \\ r_{i2} \\ \vdots \\ r_{n_i} \end{bmatrix} = \begin{bmatrix} r_{i11} & r_{i12} & r_{i13} & r_{i14} \\ r_{i21} & r_{i22} & r_{i23} & r_{i24} \\ \vdots & \vdots & \vdots & \vdots \\ r_{in_i 1} & r_{in_i 2} & r_{in_i 3} & r_{in_i 4} \end{bmatrix} \quad (17)$$

If the weight of q in r_{ij} is the biggest, $r_{ijq} = \max(r_{ij1}, r_{ij2}, r_{ij3}, r_{ij4})$, then indices V_{ij} belong to the grey cluster of q

6) Synthetic Appraisal

For evaluation indices V_i , make a synthetic evaluation, recording its synthetic evaluation result B_i , then:

$$B_i = A_i \cdot R_i = (b_{i1}, b_{i2}, b_{i3}, b_{i4})$$

According to synthetic evaluation result B_i of V_i , the grey appraisal weight coefficient matrix of appraisal grey cluster to index U , is supposed as R :

$$R = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \quad (18)$$

Then, for evaluation indices U , we make a synthetic evaluation, its synthetic evaluation result records is:

$$B = A \cdot R = (b_1, b_2, b_3, b_4) \quad (19)$$

According to synthetic evaluation result B , Under the principle of the largest belonging the gray grade of information systems security evaluated is determined.

III. APPLICATION CASE

A Example1Describe

Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization. Risk assessment is the first process in the risk management methodology. In order to verify the validity of proposed model, we select a number of information systems (IS) to carry out risk assessment. Using this method to classification grade of IS security, and the results were compared with the actual situation. The indexes reflecting Development process, Funding, Scope, Relationship Management, Scheduling, Sponsorship Ownership, External Dependencies, Project Management, Corporate Environment, Requirements, Personnel, Technology factor were adopt as shown in Fig.1.

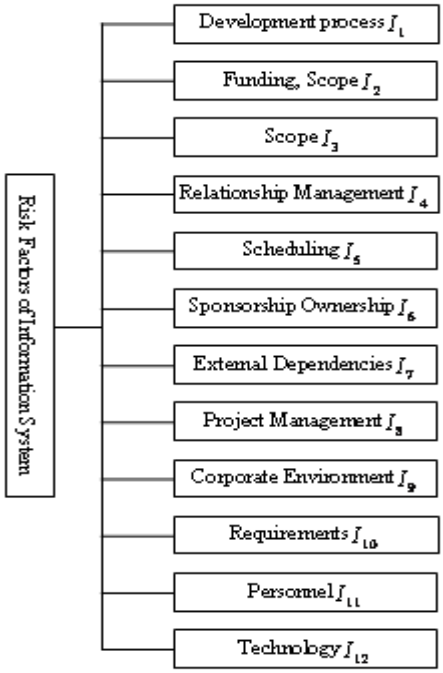


Figure 1. The security index system of IS

B Grey Relational Analysis on Information Systems Security

The data used in this study was collected from construction industry clients and consultants.

TABLE I. PRIMARY DATA

Indices	Best	Good	Qualified	Unqualified
Development process	0.23	0.22	0.37	0.18
Funding	0.23	0.20	0.32	0.25
Scope	0.22	0.12	0.40	0.26
Relationship Management	0.23	0.42	0.21	0.14

Indices	Best	Good	Qualified	Unqualified
Scheduling	0.34	0.26	0.20	0.10
Sponsorship Ownership	0.18	0.21	0.37	0.24
External Dependencies	0.25	0.35	0.29	0.11
Project Management	0.27	0.36	0.13	0.24
Corporate Environment	0.16	0.40	0.24	0.20
Requirements	0.24	0.28	0.31	0.17
Personnel	0.27	0.26	0.36	0.11
Technology factor	0.38	0.20	0.24	0.18

TABLE II. THE ANALYTICAL RESULTS

	1	2	3	4	$\min_k \Delta_i(k)$	$\max_k \Delta_i(k)$
Development process	0.15	0.04	0.12	0.13	0	0.11
Funding	0.12	0.20	0.13	0.04	0.05	0.18
Scope	0.04	0.10	0.12	0.13	0.03	0.17
Relationship Management	0.12	0.09	0.21	0.07	0.07	0.12
Scheduling	0.09	0.17	0.14	0.11	0.11	0.15
Sponsorship Ownership	0.14	0.09	0.11	0.05	0.07	0.12
External Dependencies	0	0.04	0.01	0.02	0	0.14
Project Management	0.15	0.14	0.19	0.12	0.09	0.16
Corporate Environment	0.06	0.17	0.12	0.05	0.03	0.18
Requirements	0.09	0	0.21	0.14	0	0.23
Personnel	0.04	0.17	0.13	0.16	0.12	0.18
Technology factor	0.16	0.02	0.06	0.08	0.16	0.17

TABLE III. THE RESULTS OF EXPERTS APPRAISAL AND GREY WEIGHT

Indices	1	2	3	4	5	6	r_{ij1}	r_{ij2}	r_{ij3}	r_{ij4}
q_1	3	3	3	3	3	3	0.365	0.392	0.243	0
q_2	4	2	3.5	3.5	2	3.5	0.358	0.351	0.291	0
q_3	3.5	2	3	2	3.5	3.5	0.323	0.444	0.233	0
…	…	…	..	…	…	..	…	…	…	…
q_{12}	2.5	3.5	3.5	3	4	2	0.358	0.351	0.291	0

The analytical correlation results r_i are presented below:

$$r_{10} > r_2 > r_{12} > r_1 > r_9 > r_6 > r_4 > r_8 > r_7 > r_3 > r_{11} > r_5$$

Therefore, the factors were ranked in Development process, Scheduling, Funding, Scope, Project Management, Sponsorship Ownership, Requirements, External Dependencies, Corporate Environment, Relationship Management, Personnel, Technology factor.

C Information Systems Security Assessment

1) Establishing Appraisal Indices System : The results of experts appraisal are show in Fig.2.

2) Indices System Weight with AHM :

Attribute Hierarchy Model (AHM) is derived from the Analytic Hierarchy Process (AHP) used to the mufti-criteria decision-making. The AHM is a psychometric method for classifying examinees' test item responses into a set of structured attribute patterns associated with different components from a cognitive model of task performance. Comparing with the AHP Based on the model of weight, the advantage of AHM based on the

model of game is that it's matrix does not problem of "testing of consistency."

In this paper, the authors analyzed the influential factors of the urea reactor risk and constructed the AHM. Based on the results, the weights of attributes of bids were given. The $n \times n$ pairwise comparison matrix of AHM can be got by converted AHP matrix. Then we can calculate the weighting value of the index. By using measure judgment matrix of AHM, the specific process is as follows:

Step1: construct the matrix of judgement matrix

We use the Law on 1-9 scale proposed by Satty when construct The $n \times n$ pairwise comparison matrix of AHM. The $n \times n$ pairwise comparison matrix is $A=(a_{ij})_{n\times n}$, where a_{ij} is the importance comparison with program P_i and program P_j a_{ij} meet $a_{ij} \geq 0$, $a_{ij}=1/a_{ji}(i,j=1,2,\cdots,n)$。.

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{22} \end{bmatrix} \tag{20}$$

Step2: Convert the $n \times n$ pairwise comparison

The judgment matrix to the $n \times n$ pairwise comparison measure matrix are as follow:

$$u_{ij}=\begin{cases} k/(k+1) & a_{ij}=1 \\ 0.5 & a_{ij}=1 \\ 0 & a_{ij} \\ 1/(k+1) & a_{ij}=1/k \end{cases} \tag{21}$$

Suppose u is the $n \times n$ pairwise comparison matrix.

$$u=\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{22} \end{bmatrix} \tag{22}$$

Step3: calculate the weight

$$f_i=\sum_{j=1}^{n} u_{ij}(i=1,2,\cdots,n) \tag{23}$$

$$w_i=f_i/[n(n-1)/2](i=1,2,\cdots,n) \tag{24}$$

We can get the weight vectors as follow:

$$W=(w_1,w_2,\cdots,w_n) \tag{25}$$

The index weight set of U_i and V_{ij} are determined as follow:

A=(0.42, 0.25, 0.20, 0.08, 0.05);

When, e=1

$$x_{111}=f_1(2.5)+f_1(3.5)+f_1(2.5)+f_1(2.5)+f_1(2.5)+f_1(3)$$

=0.725+0.625+0.725+0.625+0.7255+0.75=4.125

When,e=2,

$$x_{112}=f_2(3.5)+f_2(2.5)+f_2(2)+f_2(2.5)+f_2(3.5)+f_2(3)$$

=1+0.8333+0.6667+0.8333+0.8333+1=5.1666

Similarly, e=3, x_{113}=3.75; e=4, x_{114}=0 .

The objects appraised the total number belonging to various evaluation grey cluster is

x_{11}=4.125+5.1666+3.75=13.0416

e=1, r_{111}=3.125/12.0316=0.260 ;

e=2, r_{112}=5.2876/12.0316=0.4395

e=3, r_{113}=3.75/12.0316=0.312;

e=4, r_{114}=0

The index weight set is:

r_{11} = *（0.326, 0.386, 0.296, 0）*

3) Synthetic Evaluation:

Grey synthetic appraisal result for this information system is as follow:

$B=A\cdot R$=（0.32562, 0.41168, 0.2582, 0）

According to the principle of the principle of the largest belonging the gray grade of teaching quality evaluated is determined as "better".

Ⅳ. CONCLUSIONS

Grey Relational Analysis is utilized to probe the extent of connections between two digits by applying the methodology of departing and scattering measurement to the actual measurement of distance and Grey Relational Analysis is an effective means of analyzing the relationship between two series. This study applies grey relational analysis to measure the similarity between the series. This paper probes the range of factors affecting construction project duration through a literature review and a pilot survey in. The main variables shall consist of Development process, Funding, Scope, Relationship Management, Scheduling, Sponsorship Ownership, External Dependencies, Project Management, Corporate Environment, Requirements, Personnel, Technology factor. Though treatment of analytical correlation analysis, the indexes were ranked in Development process, Scheduling, Funding, Scope, Project Management, Sponsorship Ownership, Requirements, External Dependencies, Corporate Environment, Relationship Management, Personnel, Technology factor

This paper combines the measures of the grey evaluation and the group-decision analytical hierarchy process to evaluate synthetically the degree of information systems (IS) security. We builds grey hierarchy evaluated mathematics model and builds general evaluation system of customer satisfaction

through condensing the evaluation indicator system. As the evaluation work of the first two items of the evaluation targets which are established by analysis has been basically finished, the using quality is the only major task in the work of the unified check and also the hard point of grading evaluation. Focusing on the sub-issues, the gray comprehensive evaluation model used to evaluate the information systems security is established and the comprehensive evaluation results of the teaching quality is obtained; finally, we will be very easy to get a comprehensive information systems security evaluation by combining the early results of other evaluation. Thus the assessment of information systems security will be more objective. The quantitative and procedural way of evaluating the using quality of the information systems security is easy for programming and serves for the real institutions managers for decision-making in the management. This paper finally makes the security of information systems objectives are effectively controlled. It is approved by instance: we can get the good affection by using grey hierarchy evaluation method.

ACKNOWLEDGMENT

This paper is partially supported by Program for New Century Excellent Talents in University (NCET-08-0345), National Natural Science Foundation of China (60874105), Shanghai Rising-Star Program (09QA1402900), Aviation Science Foundation 20070511001), and the Chenxing Scholarship Youth Found of Shanghai Jiao Tong University (T241460612).

REFERENCES

[1] National State Auditors Association and the U. S. General Accounting Office. Management Planning Guide for Information Systems Security Auditing. December 10, 2001.
[2] Information Security Assessment. www.iss.net.
[3] Job Asheri Chaula, Louise Yngstrom, and Stewart Kowalski. Security metrics and evaluation of information systems security.
[4] H.Iranmanesh, S.Nazari Shirkouhi, and M. R. Skandari. Risk Evaluation of Information Technology Projects Based on Fuzzy Analytic Hierarchal Process World Academy of Science, Engineering and Technology 40, 2008.351-357.
[5] CCIMB-2004-01-003, 2004, Common Criteria for Information Technology Security Evaluation: Security assurance requirements and Protection Profiles Version2.2
[6] Common Methodology for Information Technology Security Evaluation Methodology Version 21999,
[7] Olegas Vasilecas, Algis Saulis, Saulius Dereskeviius. Evaluation of information systems procurement.
[8] Wu, H.H., "The Introduction of Grey Analysis," Gauli Publishing Co., Taipei, 1996.
[9] Jie Cui, Yao-Guo Dang and Si-Feng Liu. Study on Novel Property of Some Grey Relational Analysis Models. ISSN 1028-9488 Journal of Grey System, Vol. 12, No. 2, (2009) 83-88.
[10] Deng, J., Introduction to Grey System Theory. 1998
[11] Chin-Tsai Lin. Hierarchical Clustering Analysis Based on Grey Relation Grade. Information and Management Sciences, Vol. 16, No. 1, March,,2005.
[12] Wong, C. C. and Lai, H. R., A new grey relational measurement, The Journal of Grey System, Vol.12, No.4, pp.341-346, 2000.
[13] Hsia, K. H. and Wu, J. H., A study on the data preprocessing in grey relation analysis, Journal of Chinese Grey System, Vol.1, pp.47-53, 1998.
[14] Deng, J. Introduction to grey system theory. The Journal of Grey System, 1,1-24.1989
[15] G.W.Wei, Y.WEI, model of grey relational analysis for interval multiple attribute decision making with preference information o alternatives, Journal of Management Science, No.1,pp.158-162.2008.
[16] Z.X.LI, W.J.Niu, theory and application of system multi-level grey entropy optimization, Systems Engineering-Theory &Practice, No.8,pp.43-49.2007.
[17] Y.L.Ma, J.Y.Wang. Comprehensive evaluation of grey correlation analysis based on ideal plans, Systems Engineering and Electronics, No.24,pp.51-54.2002.
[18] Sun W., Guo H.(1994) Grey Assessment for Appraising the Stage of Tea Seedling, The Journal of Grey System, Vol6. No4, p.289-296. Tsai L. L., Chiu S. W., and Wu H. J.(2002)
[19] Thomas L. Saaty "Response to Holder's Comments on the Analytic Hierarchy Process" The Journal of the Operational Research Society, Vol. 42, No. 10 (Oct., 1991), pp. 909-914
[20] Holder, R.D., Some Comment on the Analytic Hierarchy Process, Journal of the Operational Research Society, 1990, 41, 11 1073-1076.
[21] A Study of Grey Relational Analysis for Synthetic Evaluation of Eutrophic Feitsui Reservoir, The Journal of Grey System, Vol. 14, No. 3, p.251-258.
[22] Gregory S. Sadesky.Standard Setting Using the Attribute Hierarchy Model Matthew M. Gushta American Institutes for Research
[23] J.L.Deng.Basic method of gray system[M] Huazhong University of Science and Technology Publishing House.1981.
[24] J.Z.Yuan.Gray System Theory and Its Application.[M] Science Press.1991.

Structural Property Analysis of Petri Net Synthesis Shared pp Subnet

Chuanliang Xia
School of Computer Science and Technology, Shandong Jianzhu University, Jinan, China
State Key Laboratory of Computer Science, Institute of Software, Academy of Sciences, Beijing, China
chuanliang_xia@126.com

***Abstract*—Petri net synthesis can avoid the state exploration problem by guaranteeing the correctness in the Petri net while incrementally expanding the net. This paper proposes the conditions imposed on a synthesis shared pp subnet under which the following structural properties will be preserved: structural boundedness, structural liveness, conservativeness, repetitiveness, consistency, P-invariant, T-invariant and fairness. Such results release the designer's burden for having to provide different methods for individual properties.**

***Index Terms*—Petri nets; synthesis; property analysis; structural boundedness; structural liveness**

I. INTRODUCTION

Petri nets are well known for their graphical and analytical capabilities in specification and verification, especially for concurrent systems. Many properties can be analytically defined and many techniques are available for development and verification. In particular, the approach based on property-preserving transformations will be described in more detail in this paper.

Usually, a design may be subject to many transformations, such as synthesis, refinement, reduction, etc. A transformation may be used for system generation or system verification. For the former, a transformation creates a needed and 'permanent' modification on a design. For the latter, a transformation is purely temporary so that verification may proceed more easily under the transformed specification. Naturally, for both purposes, it is important that a transformation should not destroy or create those properties under investigation.

In the literature, there exist three popular transformations, namely synthesis, refinement, and reduction. Synthesis is an important transformation among them. Different variations of the synthesis problem have been studied in the past [1]. Most of the efforts have been devoted to the decidability problem, i.e. questioning about of the existence of a Petri net with a specified behavior. An exception is in [2] where polynomial algorithms for the synthesis of bounded nets were presented. Desel [3] reduced the synthesis problem to the calculation of the subnet of minimal regions.

A kind of ST-net was defined in [4], under some conditions, these ST-nets can be used to model some systems. Franceschinis [5] presented the application of a compositional modeling methodology to the re-engineering of stochastic Well Formed Net (SWN) models of a contact center, the advantages are that this approach, based on the definition of classes and instances of submodels, can provide to the application of SWN to complex case studies. A module synthesis method of EN-system was presented in [6]. An F-robust net and a non-disturbing net were merged into a net [7], and the liveness is preserved. A Petri net based approach was presented in [8] to solve the digital system's high-level synthesis problem. The refinement and abstract representation method of Petri net is proposed [9], which is the key method to ensure the synthesis net preserving the well-behaved properties. Xia [10] investigated property preservations of a kind of synthesis Petri net. With some constrains, liveness and boundedness are preserved after merging some sets of subnets. A feedback control synthesis method presented in [11] provides a systematic and easily implementable tool for specialist in DES field. This method has the advantage of being applicable to both safe and non-safe PN models. Lin [12] presented a class of synthesis nets and proved that the synthesis AC systems can keep some good properties of the subsystems, such as liveness, boundedness and homestate. An approach for modeling Web service composition by Petri nets which is based on OWL-S is proposed [13]. By this approach, the boundedness and liveness properties of Petri net models are analyzed for guaranteeing the correctness of the composite Web service. An overall system Petri net model is obtained via synthesis of the individual modules satisfying system features (production rates, buffer capacities, machine expected up, down or idle time) [14].

Bergenthum [15] showed that VipTool can synthesize Petri nets from partially ordered runs and explained how VipTool including the synthesis feature can be used for a stepwise and iterative formalization and validation procedure for business process Petri net models. Kindler [16] presented mining and synthesis algorithms, which derive a Petri net model of a business process from a versioning log of a document management system. Carmona [17] presented an algorithm for the synthesis of bounded Petri nets from transition systems. This algorithm bas been implemented in a tool.

Xia [18] investigated one type of transformations and its property-preserving approach for verification. A kind of sharing Single-Link subnet synthesis method is proposed. Conditions of structural liveness preservation

doi:10.4304/jcp.7.1.292-300

of the synthesis net are proposed. In order to study property preservation of the synthesis net, this paper investigates another type of transformations and its property-preserving approach for verification. A kind of sharing pp subnet synthesis method is proposed. Under some conditions the following structural properties will be preserved: structural boundedness, structural liveness, conservativeness, repetitiveness, P-invariant, T-invariant and fairness.

This paper is organized as below. Section Ⅱ presents basic definitions. Section Ⅲ investigates structural property preservation of the ordinary synthesis net. Structural liveness preservation is studied in Section Ⅳ. Section Ⅴ gives an application example. Section Ⅵ concludes this paper.

II. BASIC DEFINITIONS

In this section we will quickly review key definitions. A more general discussion on Petri nets can be found in [20].

Definition 2.1 Let $N=(P,T;F,W)$ be a net, $\sum=(N,M_0)$ be a net system,

(1) Transition $t\in T$ is said to be enabled under M, iff $\forall p\in {}^\bullet t, M(p)\geq W(p,t)$, namely $M[t>$.

(2) If t is enabled under M, t can be fired. M can be transformed to M'.

Definition 2.2 Let $\sum=(N,M_0)$ be a net system.

(1) Transition $t\in T$ is live, if $\forall M\in R(M_0)$, $\exists M'\in R(M)$, $M'[t>$.

(2) $\sum$ is live, if $\forall t\in T$, t is live.

(3) $\sum$ is structural live, if there exists M_0, such that (N,M_0) is live.

Definition 2.3 Let $N=(P,T;F,W)$ and $N_0=(P_0,T_0;F_0,W_0)$ be two nets. If

(1) $P_0\subset P, T_0\subset T$ and $P_0\neq\phi, T_0\neq\phi$,

(2) $F_0=F\cap((P_0\times T_0)\cup(T_0\times P_0))$,

then N_0 is said to be a subnet of N.

Definition 2.4 A net $N_0=(P_0,T_0;F_0,W_0)$ is said to be a pp-type subnet of $N=(P,T;F,W)$ iff,

(1) N_0 is a subnet of N, (2) ${}^\bullet T_0\cup T_0^\bullet\subseteq P_0$,

(3) N_0 is connected, $\{p_i,p_o\}\subseteq P_0$ and p_i is the only input place of N_0, p_o is the only output place of N_0.

Supposition 2.1 A pp-type subnet satisfies:

(1) p_i is the only place which can contain the initial marking (token(s)).

(2) In a process (tokens from outside flow into p_i, pass N_0 and then flow out from p_o), the number of tokens flowing into p_i is equal to the number of tokens flowing out from p_o.

Definition 2.5 Suppose $N_1=(P_1,T_1;F_1,W_1)$ and $N_2=(P_2,T_2;F_2,W_2)$ are two Petri nets, $P_1\cap P_2=P_0\neq\phi$, $T_1\cap T_2=\phi$. $N=(P,T;F,W)$ is said to be a sharing synthesis net of N_1 and N_2, if $P=P_1\cup P_2$, $T=T_1\cup T_2$ and $F=F_1\cup F_2$.

Definition 2.6 Suppose M_{10} is the initial marking of N_1, M_{20} is the initial marking of N_2, $\forall p\in P_0$, $M_{10}(p)=M_{20}(p)$, then the initial marking of the synchronous net N is $M_0(p)=\begin{cases}M_{10}(p) & p\in P_1\\ M_{20}(p) & p\in P_2\end{cases}$.

The corresponding net system are $\sum_1=(N_1,M_{10})$, $\sum_2=(N_2,M_{20})$ and $\sum=(N,M_0)$. $\sum$ is said to be the sharing synthesis net system of $\sum_1$ and $\sum_2$.

Definition 2.7 Suppose $N_1=(P_1,T_1;F_1,W_1)$ and $N_2=(P_2,T_2;F_2,W_2)$ are two Petri nets. $N=(P,T;F,W)$ is said to be a synthesis net of N_1 and N_2 shared pp-type subnet, if the following conditions are satisfied,

(1) $P_0=P_1\cap P_2\neq\phi$, $T_0=T_1\cap T_2\neq\phi$,

(2) $P=P_1\cup P_2$, $T=T_1\cup T_2$, $F=F_1\cup F_2$,

(3) N_1 and N_2 shared pp-type subnet set N_0, where $N_0=\{N_{pp1},N_{pp2},...,N_{ppk}\}$, $N_{ppi}(i=1,2,...,k)$ are pp-type subnets.

Definition 2.8 Suppose $\sum_1=(N_1,M_{10})$ and $\sum_2=(N_2,M_{20})$ are two Petri net systems. If $\sum=(N,M_0)$ satisfies,

(1) N is the synthesis net of N_1 and N_2 shared pp-type subnets,

(2) $\forall p\in P_0$, $M_{10}(p)=M_{20}(p)$, M_0 is defined as below, $M_0(p)=\begin{cases}M_{10}(p) & p\in P_1\\ M_{20}(p) & p\in P_2\end{cases}$, then $\sum$ is said to be the synthesis net system of $\sum_1$ and $\sum_2$ shared pp-type subnets.

Definition 2.9 N is said to be structural bounded, if there exists n-dimension positive integer row vector Y such that $YA\leq 0$ (where A is the $n\times m$ incidence matrix of N).

Definition 2.10 N is said to be conservative, if there exists n-dimension positive integer row vector Y such that $YA=0$.

Definition 2.11 If there exists n-dimension positive integer row vector Y such that $YA = 0$, Y is said to be a P-invariant of N.

Definition 2.12 N is said to be repetitive, if there exists m-dimension positive integer row vector X such that $AX^{\tau} \geq 0$ (where A is the $n \times m$ incidence matrix of N, X^{τ} is the transpose of X).

Definition 2.13 N is said to be consistent, if there exists m-dimension positive integer row vector X such that $AX^{\tau} = 0$.

Definition 2.14 If there exists m-dimension positive integer row vector X such that $AX^{\tau} = 0$, X is said to be a T-invariant of N

Definition 2.15 N is said to be fair iff,

(1) For every n-dimension non-zero non-negative integer row vector X, if $A^{\tau}X = 0$, then each component of X is positive.

(2) For every two n-dimension non-zero non-negative integer row vector X_1 and X_2, if $A^{\tau}X_1 \geq 0$ and $A^{\tau}X_2 \geq 0$, then X_1 and X_2 are linearly dependent.

III. Structural Property Analysis of Ordinary Petri Synthesis Net

In this section, we investigate structural property (such as structural boundedness, conservativeness, P-invariant, repetitiveness, T-invariant and fairness) preservation of the ordinary synthesis net.

Let $N_i = (P_i, T_i; F_i, W_i)(i = 1,2)$ be two Petri nets, where $|P_i| = m_i, |T_i| = n_i$. N_1 and N_2 share a pp-type subnet $N_0 = (P_0, T_0; F_0, W_0)$ (where $|P_0| = m_0$, $|T_0| = n_0$). The synthesis net is $N = (P, T; F, W)$ (where $|P| = m$, $|T| = n$).

For the sake of convenience, we reset the P set and T set of N_1 and N_2 as below.

$$P_1 = \{p_1, p_2, ..., p_{m0}, p_{m0+1,1}, p_{m0+2,1}, ..., P_{m1}\},$$

$$P_2 = \{p_1, p_2, ..., p_{m0}, p_{m0+1,2}, p_{m0+2,2}, ..., p_{m2)}$$

$$T_1 = \{t_1, t_2, ..., t_{n0}, t_{n0+1,1}, t_{n0+2,1}, ..., t_{n1}\},$$

$$T_2 = \{t_1, t_2, ..., t_{n0}, t_{n0+1,2}, t_{n0+2,2}, ..., t_{n2}\}.$$

By Definition 2.4, A_1 ($m_1 \times n_1$ incidence matrix of N_1) and A_2 ($m_2 \times n_2$ incidence matrix of N_2) are as below.

$$A_1 = \begin{array}{c} \\ P_0 \\ P_1\text{-}P_0 \end{array} \begin{array}{c} \begin{array}{cc} T_0 & T_1\text{-}T_0 \end{array} \\ \begin{bmatrix} A_{11} & A_{31} \\ 0 & A_{41} \end{bmatrix} \end{array}$$

$$A_2 = \begin{array}{c} \\ P_0 \\ P_2\text{-}P_0 \end{array} \begin{array}{c} \begin{array}{cc} T_0 & T_2\text{-}T_0 \end{array} \\ \begin{bmatrix} A_{11} & A_{32} \\ 0 & A_{42} \end{bmatrix} \end{array}$$

Where A_{11} is a $m_0 \times n_0$ matrix; A_{41} is a $(m_1 - m_0) \times (n_1 - n_0)$ matrix; A_{42} is a $(m_2 - m_0) \times (n_2 - n_0)$ matrix.

The incidence matrix of the synthesis net N is

$$A = \begin{array}{c} \\ P_0 \\ P_1\text{-}P_0 \\ P_2\text{-}P_0 \end{array} \begin{array}{c} \begin{array}{ccc} T_0 & T_1\text{-}T_0 & T_2\text{-}T_0 \end{array} \\ \begin{bmatrix} A_{11}+A_{11} & A_{31} & A_{32} \\ 0 & A_{41} & 0 \\ 0 & 0 & A_{42} \end{bmatrix} \end{array},$$

where $|P| = m_1 + m_2 - m_0$, $|T| = n_1 + n_2 - n_0$;

$P = \{p_1, p_2, ..., p_{m0}, p_{m0+1,1}, ..., p_{m1}, p_{m0+1,2}, ..., p_{m2}\}$,

$T = \{t_1, t_2, ..., t_{n0}, t_{n0+1,1}, ..., t_{n1}, t_{n0+1,2}, ..., t_{n2}\}$.

Let $X_j = [X_{1j}^{\tau} \quad X_{2j}^{\tau}]^{\tau}$ ($j = 1,2$) be m_j dimension vectors, $Y_j = [Y_{1j}^{\tau} \quad Y_{2j}^{\tau}]^{\tau}$ ($j = 1,2$) be $n_j (j = 1,2)$ dimension vectors, where X_{1j} ($j = 1,2$) are m_0 dimension vectors, Y_{1j} ($j = 1,2$) are n_0 dimension vectors.

Theorem 3.1 Suppose that N_1 and N_2 are two structural bounded Petri nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. If there exists positive integer $k > 0$ such that $Y_{11} = kY_{12}$, then N is structural bounded.

Proof. Since N_1 and N_2 are structural bounded, by Definition 2.9, there exists positive integer vector $Y_1 = [Y_{11}^{\tau} \quad Y_{21}^{\tau}]^{\tau}$ and $Y_2 = [Y_{12}^{\tau} \quad Y_{22}^{\tau}]^{\tau}$ such that

$$A_1 Y_1 = \begin{bmatrix} A_{11} & A_{31} \\ 0 & A_{41} \end{bmatrix} \begin{bmatrix} Y_{11} \\ Y_{21} \end{bmatrix} = \begin{bmatrix} A_{11}Y_{11} + A_{31}Y_{21} \\ A_{41}Y_{21} \end{bmatrix} \leq 0,$$

$$A_2 Y_2 = \begin{bmatrix} A_{11} & A_{32} \\ 0 & A_{42} \end{bmatrix} \begin{bmatrix} Y_{12} \\ Y_{22} \end{bmatrix} = \begin{bmatrix} A_{11}Y_{12} + A_{32}Y_{22} \\ A_{42}Y_{22} \end{bmatrix} \leq 0.$$

Let $Y = \begin{bmatrix} Y_{11}^{\tau} & Y_{21}^{\tau} & kY_{22}^{\tau} \end{bmatrix}^{\tau}$, then

$$AY = \begin{bmatrix} A_{11} + A_{11} & A_{31} & A_{32} \\ 0 & A_{41} & 0 \\ 0 & 0 & A_{42} \end{bmatrix} \begin{bmatrix} Y_{11} \\ Y_{21} \\ kY_{22} \end{bmatrix}$$

$$= \begin{bmatrix} (A_{11}Y_{11} + A_{31}Y_{21}) + k(A_{11}Y_{12} + A_{32}Y_{22}) \\ A_{41}Y_{21} \\ kA_{42}Y_{22} \end{bmatrix} \le 0$$

By Definition 2.9, N is structural bounded.

Theorem 3.2 Suppose that N_1 and N_2 are two conservative Petri nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. If there exists positive integer $k > 0$ such that $Y_{11} = kY_{12}$, then N is conservative.

Theorem 3.3 Suppose that N_1 and N_2 are two Petri nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. Y_1 is a T-invariant of N_1 . Y_2 is a T-invariant of N_2 . If there exists positive integer $k > 0$ such that $Y_{11} = kY_{12}$, then $Y = [Y_{11}^{\tau}, Y_{21}^{\tau}, kY_{22}^{\tau}]^{\tau}$ is a T-invariant of N .

Theorem 3.4 Suppose that N_1 and N_2 are two repetitive Petri nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. If there exists positive integer $k > 0$ such that $X_{11} = kX_{12}$, then N is repetitive.

Proof. Since N_1 and N_2 are two repetitive Petri nets, by Definition 2.12, there exists positive vector $X_1 = [X_{11}^{\tau} \quad X_{21}^{\tau}]^{\tau}$ and $X_2 = [X_{12}^{\tau} \quad X_{22}^{\tau}]^{\tau}$ such that

$$A_1^{\tau} X_1 = \begin{bmatrix} A_{11}^{\tau} & 0 \\ A_{31}^{\tau} & A_{41}^{\tau} \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}^{\tau} X_{11} \\ A_{31}^{\tau} X_{11} + A_{41}^{\tau} X_{21} \end{bmatrix} \ge 0,$$

$$A_2^{\tau} X_1 = \begin{bmatrix} A_{11}^{\tau} & 0 \\ A_{32}^{\tau} & A_{42}^{\tau} \end{bmatrix} \begin{bmatrix} X_{12} \\ X_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}^{\tau} X_{12} \\ A_{32}^{\tau} X_{12} + A_{42}^{\tau} X_{22} \end{bmatrix} \ge 0.$$

Let $X = \begin{bmatrix} X_{11}^{\tau} & X_{21}^{\tau} & kX_{22}^{\tau} \end{bmatrix}^{\tau}$,

we have

$$A^{\tau} X = \begin{bmatrix} A_{11}^{\tau} + A_{11}^{\tau} & 0 & 0 \\ A_{31}^{\tau} & A_{41}^{\tau} & 0 \\ A_{32}^{\tau} & 0 & A_{42}^{\tau} \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \\ kX_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}^{\tau} X_{11} + A_{11}^{\tau} X_{11} \\ A_{31}^{\tau} X_{11} + A_{41}^{\tau} X_{21} \\ A_{32}^{\tau} X_{11} + A_{42}^{\tau} kX_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}^{\tau} X_{11} + kA_{11}^{\tau} X_{12} \\ A_{31}^{\tau} X_{11} + A_{41}^{\tau} X_{21} \\ k(A_{32}^{\tau} X_{12} + A_{42}^{\tau} X_{22}) \end{bmatrix} \ge 0.$$

By Definition 2.12, N is repetitive.

Theorem 3.5 Suppose that N_1 and N_2 are two consistent Petri nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. If there exists positive integer $k > 0$ such that $X_{11} = kX_{12}$, then N is consistent.

Theorem 3.6 Suppose that N_1 and N_2 are two Petri nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. X_1 is a P-invariant of N_1 . X_2 is a P-invariant of N_2 . If there exists positive integer $k > 0$ such that $X_{11} = kX_{12}$, then $X = [X_{11}^{\tau}, Y_{21}^{\tau}, kX_{22}^{\tau}]^{\tau}$ is a P-invariant of N .

Theorem 3.7 Suppose that N_1 and N_2 are two fair Petri nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. Then N is fair.

Proof. (1) Since N_1 and N_2 are two fair Petri nets, by Definition 2.15, if every n_i ($i = 1,2$) dimension non-zero non-negative integer vector X_i satisfies $A_i^{\tau} X_i \ge 0$, then $X_i > 0$, i.e.,

If $A_1^{\tau} X_1 = \begin{bmatrix} A_{11}^{\tau} & 0 \\ A_{31}^{\tau} & A_{41}^{\tau} \end{bmatrix} \begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix}$

$$= \begin{bmatrix} A_{11}^{\tau} X_{11} \\ A_{31}^{\tau} X_{11} + A_{41}^{\tau} X_{21} \end{bmatrix} \ge 0, \text{ then } X_{11} > 0, X_{21} > 0.$$

If $A_2^{\tau} X_1 = \begin{bmatrix} A_{11}^{\tau} & 0 \\ A_{32}^{\tau} & A_{42}^{\tau} \end{bmatrix} \begin{bmatrix} X_{12} \\ X_{22} \end{bmatrix}$

$$= \begin{bmatrix} A_{11}^{\tau} X_{12} \\ A_{32}^{\tau} X_{12} + A_{42}^{\tau} X_{22} \end{bmatrix} \ge 0, \text{ then } X_{12} > 0, X_{22} > 0.$$

Suppose $X = [X_{10}^{\tau}, X_{20}^{\tau}, X_{30}^{\tau}]^{\tau}$ is a n-dimension non-zero non-negative integer vector, where X_{10} is a n_0-dimension vector, X_{20} is a $(n_1 - n_0)$-dimension vector, X_{30} is a $(n_2 - n_0)$-dimension vector.

Since $A^{\tau}X=\begin{bmatrix} A_{11}^{\tau}+A_{11}^{\tau} & 0 & 0 \\ A_{31}^{\tau} & A_{41}^{\tau} & 0 \\ A_{32}^{\tau} & 0 & A_{42}^{\tau} \end{bmatrix}\begin{bmatrix} X_{10} \\ X_{20} \\ X_{30} \end{bmatrix}$

$=\begin{bmatrix} A_{11}^{\tau}X_{10}+A_{11}^{\tau}X_{10} \\ X_{31}^{\tau}X_{10}+A_{41}^{\tau}X_{20} \\ A_{32}^{\tau}X_{10}+A_{42}^{\tau}X_{30} \end{bmatrix}=\begin{bmatrix} 2A_{11}^{\tau}X_{10} \\ A_{31}^{\tau}X_{10}+A_{41}^{\tau}X_{20} \\ A_{32}^{\tau}X_{10}+A_{42}^{\tau}X_{30} \end{bmatrix}\geq 0.$

Obviously, $X_{10}>0, X_{20}>0, X_{30}>0$, i.e., $X>0$.

(2) If $A_i^{\tau}X_i^{(1)}\geq 0$ and $A_i^{\tau}X_i^{(2)}\geq 0$, where $X_i^{(1)}$ and $X_i^{(2)}$ ($i=1,2$) are non-zero non-negative integer vectors, by Definition 2.15, $X_i^{(1)}$ and $X_i^{(2)}$ are linearly dependent. For any two non-zero non-negative integer vector $X^{(1)}$ and $X^{(2)}$, if $A^{\tau}X^{(1)}\geq 0$ and $A^{\tau}X^{(2)}\geq 0$, then $X^{(1)}$ and $X^{(2)}$ are linearly dependent (the proof is similar to that of (1)).

By Definition 2.15, N is fair.

Ⅳ. Structural Liveness Preservation of Synthesis Net

In this section, some conditions of structural liveness preservation of synthesis net are proposed. This is important to property analysis of complex nets.

Definition 4.1 pp-type subnet refinement operation: $N'=(P',T';F',W')$ is obtained from Petri net $N=(P,T;F,W)$ by using a pp-type subnet $N_{pp}=(P_{pp},T_{pp};F_{pp},W_{pp})$ to replace $\tilde{p}$ ($\tilde{p}\in P$), where

(1) $P'=(P-\{\tilde{p}\})\cup P_{pp}$, (2) $T'=T\cup T_{pp}$,

(3) $F'=F\cup\{(t,p_i)\mid t\in {}^{\bullet}\tilde{p}\}\cup F_{pp}\cup\{(p_0,t)\mid t\in \tilde{p}^{\bullet}\}$

$-\{(t,\tilde{p})\mid t\in {}^{\bullet}\tilde{p}\}-\{(\tilde{p},t)\mid t\in \tilde{p}^{\bullet}\}$.

Definition 4.2 A net $(\overline{N}_{pp},\overline{M}_{ppo})$ is said to be a pp-type closed net if we add a transition set $T_{tpp}=\{t_{pp}\mid t_{pp}$ corresponding to $t\in\tilde{p}^{\bullet}\}$ and arc set $\{(p_o,t_{pp}),(t_{pp},p_i)\mid t\in T_{tpp}\}$ to (N_{pp},M_{pp0}), and preserve the marking of (N_{pp},M_{pp0}).

Definition 4.3 (N',M_0') obtained from (N,M_0) by pp-type subnet refinement operation comprises net N' and marking M_0', where

$$M_0'=\begin{cases}(M_{(P\setminus\tilde{p})0},\theta_{pp}) & M_0(\tilde{p})=0\\ (M_{(P\setminus\tilde{p})0},M_{pp0}) & M_0(\tilde{p})>0\end{cases}$$

($M_{(P\setminus\tilde{p})}$ is obtained from M by deleted the vector corresponding to $\tilde{p}$, θ_{pp} is 0-vector of M_{pp}).

Definition 4.4 pp-type subnet abstract operation: $N'=(P',T';F',W')$ is obtained from Petri net $N=(P,T;F,W)$ by using a place $\tilde{p}$ to replace a pp-type subnet $N_{pp}=(P_{pp},T_{pp};F_{pp},W_{pp})$, where (1) $P'=(P-P_{pp})\cup\{\tilde{p}\}$, (2) $T'=T-T_{pp}$,

(3) $F'=(F-F_{pp}-\{(t,p_i)\mid t\in {}^{\bullet}p_i\}-\{(p_o,t)\mid t\in p_o^{\bullet}\})$

$\cup\{(t,\tilde{p})\mid t\in {}^{\bullet}p_i\}\cup\{(\tilde{p},t)\mid t\in p_o^{\bullet}\}$.

Definition 4.5 (N',M_0') obtained by pp-type subnet abstract operation comprise net N' and marking M_0', where $M_0'=\begin{cases}(M_{(P\setminus pp)0},0), & M_0(p_i)=0,\\ (M_{(P\setminus pp)0},M(p_i)), & M_0(p_i)>0.\end{cases}$

($M_{(P\setminus pp)}$ is obtained from M by deleted the vector corresponding to P_{pp}, and $M_0'(\tilde{p})=M_0(p_i)$).

Lemma 4.1 Suppose that N' is obtained from N by pp-type refinement operation. N' is structural bounded iff N and $\overline{N}_{pp}$ are structural bounded.

Proof. ($\Leftarrow$)([19]).

($\Rightarrow$) By the reduction to absurdity method, it is easy to prove.

Lemma 4.2 Suppose that N' is obtained from N by pp-type refinement operation. N' is structural live iff N and $\overline{N}_{pp}$ are structurallive .

The proof is adapted from [9].

Definition 4.5 Let N be a net,

(1) N is said to be a Free Choice net (FC), if $\forall p\in P,\mid p^{\bullet}\mid\leq 1$ or ${}^{\bullet}(p^{\bullet})=\{p\}$.

(2) N is an Asymmetric Choice net (AC), if $\forall p_1,p_2\in P, p_1^{\bullet}\cap p_2^{\bullet}\neq\phi\Rightarrow p_1^{\bullet}\subseteq p_2^{\bullet}$ or $p_2^{\bullet}\subseteq p_1^{\bullet}$.

Definition 4.6 [12] Let N be a net,

N is said to be a Decomposable Asymmetric Choice Net (DAC), iff $N=N_1\cup N_2$ and $N_1\cap N_2=\{s\}$, where $s\in P$, N_1 and N_2 are FC nets.

Definition 4.7 [12] Let N be a net,

N is said to be an Extended Decomposable Asymmetric Choice Net (EDAC), iff $N=N_1\cup N_2$ and $N_1\cap N_2=\{s_1,s_2,...,s_n\}$, where $s_i\in P$ ($i=1,2,...,n$), N_1 and N_2 are FC nets.

Lemma 4.3 [12] Suppose that N is a DAC net, $N=N_1\cup N_2$, $N_1\cap N_2=\{s\}$ where $s\in P$, N_1

and N_2 are FC nets. N is structural bounded iff N_1 and N_2 are structural bounded.

Theorem 4.1 Suppose that N_1 and N_2 are two FC nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. N is structural bounded iff N_1 and N_2 are structural bounded.

Proof. ($\Leftarrow$)Suppose the pp-type subnet is N_0. Let N_1' and N_2' be obtained from N_1 and N_2 by pp-type abstract operation, respectively. N_0 is replaced by $\widetilde{p}$. Since N_1 and N_2 are two FC nets, then N_1' and N_2' are two FC nets. For N_1' and N_2', $N'=N_1'\cup N_2'$, $N_1'\cap N_2'=\{\widetilde{p}\}$, by Lemma 4.3, N' is structural bounded. By Lemma 4.1, the closed net $\overline{N}_0$ is structural bounded. Let N be obtained from N' by pp-type refinement operation. In the process $\widetilde{p}$ should be replaced by N_0. Since N' and $\overline{N}_0$ are structural bounded, by Lemma 4.1, N is structural bounded.

($\Rightarrow$) Without losing of generality, suppose N_1 is not structural bounded, by Lemma 4.1, N_1' is not structural bounded. Obviously, N' is not structural bounded. By Lemma 4.1, N is not structural bounded. This contradicts with the fact that N is structural bounded.

Lemma 4.4 [12] Suppose that N is a DAC net, $N=N_1\cup N_2$, $N_1\cap N_2=\{s\}$ where $s\in P$, N_1 and N_2 are FC nets. N is structural bounded and structural live iff N_1 and N_2 are structural bounded and structural live.

Theorem 4.2 Suppose that N_1 and N_2 are two FC nets, N is the synthesis net of N_1 and N_2 which shared a pp-type subnet. N is structural bounded and structural live iff N_1 and N_2 are structural bounded and structural live.

Proof. ($\Leftarrow$)Suppose the pp-type subnet is N_0. Let N_1' and N_2' be obtained from N_1 and N_2 by pp-type abstract operation, respectively. N_0 is replaced by $\widetilde{p}$. Since N_1 and N_2 are two FC nets, then N_1' and N_2' are two FC nets. Since N_1 and N_2 are structural live, by Lemma 4.2, N_1' and N_2' are structural live. Since N_1 and N_2 are structural bounded, by Lemma 4.1, N_1' and N_2' are structural bounded. For N_1' and N_2', $N'=N_1'\cup N_2'$, $N_1'\cap N_2'=\{\widetilde{p}\}$, by Lemma 4.4, N' is structural bounded and structural live. By Lemma 4.1 and Lemma 4.2, the closed net $\overline{N}_0$ is structural bounded and structural live. Let N be obtained from N' by pp-type refinement operation. In the process $\widetilde{p}$ should be replaced by N_0. Since N' and $\overline{N}_0$ are structural bounded and structural live, by Lemma 4.1 and Lemma 4.2, N is structural bounded and structural live.

($\Rightarrow$) Without losing of generality, suppose N_1 is not structural bounded and structural live, by Lemma 4.1 and Lemma 4.2, N_1' is not structural bounded and structural live. Obviously, N' is not structural bounded and structural live. By Lemma 4.1 and Lemma 4.2, N is not structural bounded and structural live. This contradicts with the fact that N is structural bounded and structural live.

Lemma 4.5[12] Suppose that N is a AC net, $N=\bigcup_i N_i$, where N_i ($i=1,2,...,m$) are FC nets. $N_j\bigcap N_{j+1}=\{s_{j,j+1}\}$, where $s_{j,j+1}\in P$ ($j=1,2,...,m-1$). Other subnets have no intersection. If N_i ($i=1,2,...,m$) are structural live, then N is structural live.

Theorem 4.3 Suppose that N_i ($i=1,2,...,m$) are FC nets. $N_j\bigcap N_{j+1}=\{N_{0(j,j+1)}\}$, where $N_{0(j,j+1)}$ ($j=1,2,...,m-1$) are pp-type subnets. Other subnets have no intersection. If N is an AC net and N_i ($i=1,2,...,m$) are structural live, then N is structural live.

Proof. Suppose the pp-type subnets are N_{0i} ($i=1,2,...,m$). Let $N_i'(i=1,2,...,m)$ be obtained from $N_i(i=1,2,...,m)$ by pp-type abstract operation. Obviously, $N_i'(i=1,2,...,m)$ are FC nets. Since $N_i(i=1,2,...,m)$ are structural live, by Lemma 4.2, $N_i'(i=1,2,...,m)$ are structural live. Since $N_j\bigcap N_{j+1}=\{N_{0(j,j+1)}\}$ (where $N_{0(j,j+1)}$ ($j=1,2,...,m-1$) are pp-type subnets) and other subnets have no intersection, then $N_j'\bigcap N_{j+1}'=\{p_{j,j+1}\}$ (where $p_{j,j+1}\in P'$ ($j=1,2,...,m-1$)) and other subnets have no intersection. Let $N'=\bigcup_i N_i'$ ($i=1,2,...,m$). Since N is an AC net, then N' is an AC net. Since $N_i'(i=1,2,...,m)$ are structural live, by Lemma 4.5, N' is structural live. By Lemma 4.2, the closed nets $\overline{N}_{0i}(i=1,2,...,m)$ are structural live. Let N be obtained from N' by pp-type refinement operation, i.e., $p_{j,j+1}$ ($j=1,2,...,m-1$) should be replaced by $N_{0(j,j+1)}$ ($j=1,2,...,m-1$). By Lemma 4.2, N is structural live.

Lemma 4.6 [12] Suppose that $\sum = (N, M_0)$ is an EDAC net system, the corresponding subnet systems are $\sum_1 = (N_1, M_0^{N1})$ and $\sum_2 = (N_2, M_0^{N2})$. $N = N_1 \cup N_2$ and $N_1 \cap N_2 = S = \{s_1, s_2, ..., s_n\}$, where $s_i \in P$ ($i = 1,2,...,n$, $n \geq 1$). If $\sum_1$ and $\sum_2$ are live, then $\sum$ is live.

Theorem 4.4 Suppose that N is an EDAC net, the corresponding subnets are N_1 and N_2. $N = N_1 \cup N_2$ and $N_1 \cap N_2 = S = \{s_1, s_2, ..., s_n\}$, where $s_i \in P$ ($i = 1,2,...,n$, $n \geq 1$). If N_1 and N_2 are structural bounded and structural live, then N is structural live.

Theorem 4.5 Suppose that N_1 and N_2 are FC nets, $N = N_1 \cup N_2$ and $N_1 \cap N_2 = \{N_{01}, N_{02}, ..., N_{0n}\}$, where $N_{0i} (i = 1,2,...,n, n \geq 1)$ are a pp-type subnets. If N_1 and N_2 are structural bounded and structural live, then N is structural live.

Proof. Suppose the pp-type subnets are N_{0i} ($i = 1,2,...,n$). Let N_1' and N_2' be obtained from N_1 and N_2 by pp-type abstract operation, respectively. N_{0i} ($i = 1,2,...,n$) are replaced by p_i ($i = 1,2,...,n$). Obviously, N_1 and N_2 are FC nets. $N_1' \cap N_2' = \{p_1, p_2, ..., p_n\}$. Let $N' = N_1' \cup N_2'$. Since N_1 and N_2 are structural bounded and structural live, by Lemma 4.1 and Lemma 4.2, N_1', N_2' and the closed nets $\overline{N}_{0i} (i = 1,2,...,n)$ are structural bounded and structural live. Since N_1' and N_2' are structural bounded and structural live, by Theorem 4.4, N' is structural live. Let N be obtained from N' by pp-type refinement operation, i.e., p_i ($i = 1,2,...,n$) should be replaced by N_{0i} ($i = 1,2,...,n$). By Lemma 4.2, N is structural live.

V. APPLICATIONS

In reality, when several subsystems use same resources, such as robots and tools et al. and transitions of the same subnet have the same enabled time or conditions, these subsystems can be synthesized by sharing the same subnets. The structural liveness preservation guarantee the well-balanced run of the synthesis net system.

In this section we will use the synthesis method to model the system that plant_1 and plant_2 use two machines of a workstation to produce two different parts.

The model of N_1 is described in Fig.1.

The meaning of transitions and places of Fig.1 is as below.

t_1 : process raw material_1 on machine_1;

t_2 : process raw material_2 on machine_2;

t_3 : finish process on machine_1;

t_4 : finish process on machine_2;

t_5 : reprocess semi-manufactured part_1 and semi-manufactured part_2 on machine_1;

t_6 : reprocess semi-manufactured part_1 and semi-manufactured part_2 on machine_2;

t_7 : refine process part_1 on machine_1;

t_8 : refine process part_2 on machine_2;

t_9, t_{11} : transfer finished part_1;

t_{10}, t_{12} : transfer finished part_2;

t_{13} : assemble;

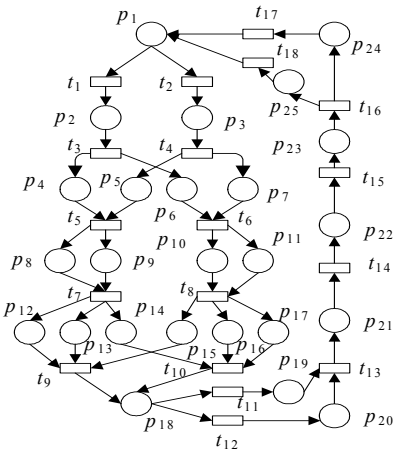


Fig. 1 Net N_1

t_{14} :sell finished product;

t_{15} : formulate production plan for the next step ;

t_{16} :prepare raw materials;

t_{17} : transfer raw material_1;

t_{18} : transfer raw material_2;

p_1 : raw material_1 and raw material_2;

p_2 : semi-manufactured part_1;

p_3 : semi-manufactured part_2 ;

p_4, p_6 : semi-manufactured part_1;

p_5, p_7 : semi-manufactured part_2;

p_8, p_{11} : machine_2 is idle;

p_9 : coarse part_1;

p_{10} : coarse part_2;

p_{12}, p_{14} : machine_1 is idle;

p_{13} : finished part_1;

p_{15}, p_{17} : machine_2 is idle;

p_{16} : finished part_2;

p_{18} : finished part_1 and finished part_2;

p_{19} :finished part_1;

p_{20} : finished part_2; p_{21} : product;

p_{22} : rest state;

p_{23} : the next step of the production plan;

p_{24} : raw material_1;

p_{25} : raw material_2;

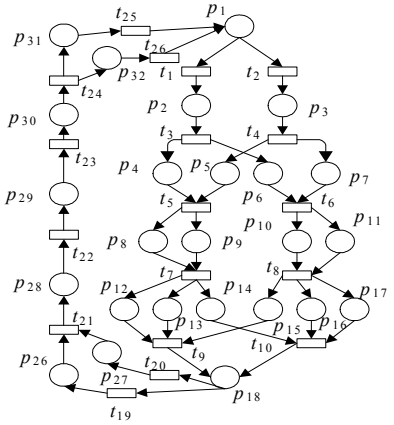


Fig. 2 Net N_2

The model of N_2 is described in Figure 2, where meaning of some transitions and places is similar to those of Fig. 1.

t_{19}: transfer finished parts_1;

t_{20} : transfer finished parts_2;

t_{21} : assemble; t_{22} : deliver product;

t_{23} : receive order; t_{24} : prepare raw materials;

t_{25} : transfer raw materials_1;

t_{26} : transfer raw materials_2;

p_{26} : finished parts_1; p_{27} : finished parts_2;

p_{28} : products; p_{29} : rest state; p_{30} : orders;

p_{31} : raw materials_1; p_{32} : raw materials_2.

Plant_1 and plant_2 use two machines of a workshop to produce two different parts, that is, these two plants share a same set of resources. The synthesis net N (Fig.3) is obtained by the synthesis of N_1 and N_2 which shared a pp-type subnet.

According to Fig.1 and Fig.2, it is easy to see that N_1 and N_2 are two structural bounded and structural live FC nets. By Theorem 4.2, the synthesis net N is a structural bounded and structural live FC net.

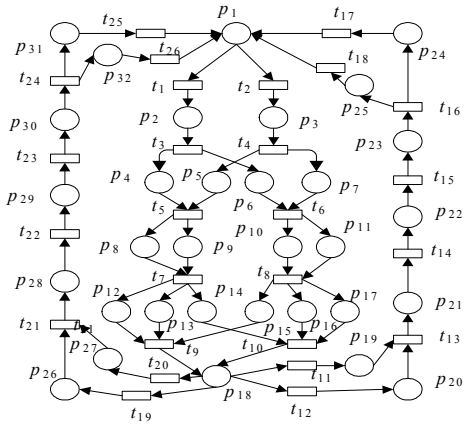


Fig. 3 Synthesis Net *N*

VI. CONCLUSIONS

In this paper we investigate structural property preservations of Petri synthesis net. A Petri net synthesis method is proposed, which is the key method to ensure the synthesis net preserving well behaved properties. Conditions of structural property preservation of ordinary Petri synthesis net are proposed. Implementation of the Petri net synthesis shared pp-type subnets can achieve resource sharing and synchronization of operations to solve the system of scheduling problems. As a consequence, this result can be applied nicely to solve some of the subsystem sharing problem in management engineering, manufacturing engineering. Further research is needed to give more general conditions to investigate other property preservations of the synthesis net.

ACKNOWLEDGMENT

This work was financially supported by the National Natural Science Foundation of China under Grant No. 60573012, 60672180 and 60721061, the National Grand Fundamental Research 973 Program of China under Grant No. 2002cb312200, and CAS Key Laboratory of Computer Sciences (SYSKF0903).

REFERENCES

[1] P. Darondeau, Synthesis and control of asynchronous and distributed systems. In: T. Basten, G. Juhás, S. K. Shukla (eds.) ACSD. IEEE Computer Society, Los Alamitos (2007) pp. 13-22.

[2] E. Badouel, L. Bernardinello, P. Darondeau, Polynomial algorithms for the synthesis of bounded nets. In: P.D. Mosses, M.I. Schwartzbach, M. Nielsen (eds.) TAPSOFT 1995. LNCS, vol. 915, pp.364-383. Springer Heidelberg (1995).

[3] J. Desel, W. Reisig, The synthesis problem of Petri nets. Acta Informatica 33(4), 297-315 (1996).

[4] Kees van Hee, Natalia Sidorova, et al. Soundness and separability of workflow nets in the stepwise refinement. In: Proc the 24th International Conference on Application and Theory of Petri Nets. Eindhoven, The Netherlands, 2003, 337-356.
[5] G. Franceschinis, M. Gribaudo, et al., Compositional modeling of complex systems: contact center scenarios in OsMoSys, Proc. 25th International Conference on Application and Theory of Petri nets, Bologna, Italy, (2004) 177-196.
[6] L. Bernardinello, C. Ferigato, et al. Towards modular synthesis of EN systems. In: B. Caillaud, et al. (eds.) Synthesis and Control of Discrete Event Systems, 103-113, Kluwer Academic Publishers, 2002
[7] Y. Souissi, On liveness preservation by composition of nets via a set of places. In: Rozenberg G, ed. LNCS 483, New York: Springer-Verlag, 1990. 457-470.
[8] Victor R. L. Shen, A PN-based approach to the high-level synthesis of digital systems, INTEGRATION, yhe VLSI journal 39 (2006) 182-124.
[9] C. Xia, Analysis of properties of Petri synthesis net, Proc the 3rd International Conference on Theory and Applications of Models of Computation, LNCS 3959, (2006) 576-587.
[10] C. Xia, Property preservation by Petri net synthesis in system design, Proc. of International Technology and Environmental System Sciences, 2008, Vol. 2, 6-11.
[11] A. I. Vasiliu, A. Dideban, H. Alla, Control synthesis for manufacturing systems using non-safe Petri nets. CEAI Vol. 11, No.2, pp. 43-50, 2009.
[12] G. X. Lin, W. M. Lu, Liveness, Boundedness and HomeState of Decomposable Asymmetric Choice Nets. J. Computers, 2002,25(12):1325-1330.
[13] Z. J. Ding, J. L. Wang, C. J. Jiang, An approach for synthesis Petri nets for modeling and verifying composite web service. Journal of Information Science and Engineering 24, 1309-1328 (2008).
[14] G. J. Tsinarakis, N. C. Tsourveloudis, K. P. Valavanis, Modular Petri net based modeling, analysis, synthesis and performance evaluation of random topology dedicated production systems. Journal of Intelligent manufacturing, 16 (2005) 67-92.
[15] R. Bergenthum, J. Desel, R. Lorenz, S. Mauser, Synthesis of Petri nets from scenarios with VipTool. PETRI NETS 2008, LNCS 5062, pp. 388-398, 2008.
[16] E. Kindler, V. Rubin, W. Schäfer, Process Mining and Petri net synthesis. BPM 2006 Workshops, LNCS 4103, pp.105-116, 2006.
[17] J. Carmona, J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, A. Yakovlev, A symbolic algorithm for the synthesis of bounded Petri nets. PETRI NETS 2008, LNCS 5062, pp. 92-111, 2008.
[18] C. Xia, A Petri net synthesis method. Journal of Networks, Vol. 5, No. 6, pp, 699-707, 2010.
[19] C. Xia, Property preservation of Petri refinement net. 2009 WRI World Congress on Computer Science and Information Engineering (CSIE 2009), Vol. 5, pp.749-754, 2009.
[20] T. Murata, Petri nets: properties, analysis and applications. Proceedings of IEEE, 1989, Vol.77, pp541-580.

Applying Principal Component Analysis, Genetic Algorithm and Support Vector Machine for Risk Forecasting of General Contracting

Huawang Shi
School of Civil Engineering, Hebei University of Engineering, Handan, P.R.China
e-mail: stone21st@163.com

***Abstract*—In order to evaluate and forecast the general contracting risk, a multi-resolution approach for the price determination of real estate was present in this paper. Real samples have been classified using the novel multi-classifier, namely, support vector machine among which genetic algorithm (GA) is used to determine free parameters of support vector machine. Effects of different sampling approach, kernel functions, and parameter settings used for SVM classification are thoroughly evaluated and discussed. The experimental results indicate that the SVMG method can achieve greater accuracy than grey model, artificial neural network under the circumstance of small training data. It was also found that the predictive ability of the SVM outperformed those of some traditional pattern recognition methods for the data set used here.**

***Index Terms*—support vector machines; principal component analysis; genetic algorithm; risk forecasting; general contracting**

I. INTRODUCTION

At present, in the construction industry, it is a tendency to put into practice the general contracting in engineering project. China also enthusiastically develops general contracting. Construction projects are one-of endeavors with many unique features such as long period, complicated processes, abominable environment, financial intensity and dynamic organization structures [1,2] and such organizational and technological complexity generates enormous risks[3]. The general contracting in engineering project is subjected to the long period, numerous participants and involving government, economic, society, community, culture, technology, environment and other factors. And along with the world economy's continuous development and the project scope's increase extension, the loss caused of the risk occurring will become more great, and the effect to relevant parties will become more distinct too.

How to enhance the risk management is an important question for discussion. The contractor takes more risks under the mode of the engineering general contract. The paper starts from the relative concepts of engineering risk and the theory of the risk management. With the need for improved performance in the construction industry and increasing contractual obligations, the requirement of an effective risk management approach has never been more necessary. Risk assessment is a complex subject shrouded in vagueness and uncertainty. There are some risk assessment methods now. These risk assessment methods are all based on fuzzy set theory.

According to characteristics and risks of the general engineering, the paper brings forward the risk that the company should be pay attention to, allowing for the theory of total risk management and system and the characteristic of the mode about engineering general contract. The application model of risk management frame system is presented by analyzing the source and characteristic of the engineering risk, and the theory of the risk management. The model based on fuzzy mathematic and analytic hierarchy process(AHP) is used to evaluate risks and some advice is given. There are many factors affecting the accident risk of construction, but some of the factors are related and redundant. PCA is a powerful tool for analyzing data. The goal of PCA is to reduce the dimensionality of the data while retaining as much as possible of the variation present in the original data set. In this paper, we use principal component analysis (PCA) to reduce some related or redundant general contract factors. The paper presents advice for the general engineering contract risk management, provides reference to participation in engineering contract risk management and gives the suggestion of cultivating the ability of competition for the civil engineering general company. Those can be great helpful to develop the market for engineering general company, at the same time also can present theory and practice guidance for dealing with the risk of the project general contractors.

Genetic Algorithms (GAs), which imitate parts of the natural evolution process, were first proposed by Holland [9]. Genetic algorithm does not require a gradient of the objectiveness function as a search direction, it can automatically acquire and accumulate knowledge on search space and adaptive control the searching process, so as Gas are stochastic search approaches inspired by natural evolution that involve crossover, mutation, and evaluation of survival fitness. Gas out perform the efficiency of conventional optimization techniques in searching non-linear and non-continuous spaces, which are characterized by abstract or poorly understood expert knowledge. Furthermore, to the contrary with the standard algorithms, Gas generate at each iteration population of points that approach the optimal solution by using stochastic and not deterministic operators. As a result, the search can be deployed without being trapped in local extremes. Based on its merits, the potential of using GA in optimization techniques has been in extensively studied [3,4]. However, simple GA is difficult to apply directly

doi:10.4304/jcp.7.1.301-307

and successfully to a larger range of difficult-to-solve optimization problems.

Developed by Vapnik, SVM is the method that is receiving increasing attention with remarkable results recently. The main difference between ANN and SVM is the principle of risk minimization. ANN implements empirical risk minimization to minimize the error on the training data. However, support vector machine (SVM) implements the principle of structural risk minimization in place of experiential risk minimization, which makes it have excellent generalization ability in the situation of small sample. In addition, SVM can change a non-linear learning problem into a linear learning problem in order to reduce the algorithm complexity by using the kernel function idea present, SVM has been applied successfully to solve non-linear regression estimation problems in financial time series forecasting, bankruptcy prediction, reliability prediction, etc. In this paper, the proposed SVMG model is applied to research the forecasting problem of the ratios of key-gas in power transformer oil, among which GA is used to optimize the parameters of support vector machine, because the election of the parameters plays an important role in the performance of SVM.

This paper is organized as follows: Section 2 introduces the methodology including Principal component analysis (PCA), Genetic Algorithm and regression arithmetic of support vector machine SVM model. The foundations of support vector machines are introduced. The proposed model is presented Section 3 testifies the performance of the proposed model with the real data sets from several companies in China. Finally, the conclusion is provided in Section 4.

II. METHODOLOGY

A. Introduction to PCA

Principal component analysis (PCA) was invented in 1901 by Karl Pearson[2]. Now it is mostly used as a tool in exploratory data analysis and for making predictive models. PCA involves the calculation of the eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute. The results of a PCA are usually discussed in terms of component scores and loadings (Shaw, 2003). PCA[3-7] can be used for dimensionality reduction in a data set by retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data. However, depending on the application this may not always be the case.

Problems arise when performing recognition in a high-dimensional space (e.g., curse of dimensionality). Significant improvements can be achieved by first mapping the data into a lower-dimensionality space. The goal of PCA is to reduce the dimensionality of the data while retaining as much as possible of the variation present in the original data set.

Supposing n samples, each sample has m target factors, x_j (j = 1,2, ..., m), derived from observation values x_{ij} (i=1,2,...,n), constitute the raw data matrix $X=(x_{ij})n\times m$, shown as bellow:

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{1}$$

The target factor is often relevant, thus increasing the internal complexity of the samples. Principal component analysis is to have a correlation between a number of factors into a set of mutually independent factor of a few General methods. These will be the original general factor target factor in the overlapping information removed, to the original contains only significant difference between the target and reflect the original main target factor information purposes. That is, without changing the original data provided by the basic information on more focused and typically show the characteristics of the study. Principal component - the specific algorithm for cluster analysis are as follows.

(1) Original data will be standardized (Z-Score Standardization)

Class and quantity in order to eliminate the impact of different dimension, first of all original data on the standardization of treatment (standardized value of the post-treatment x_{ij}*

$$x_{ij}^{*}=\frac{x_{ij}-\overline{x_j}}{S_j} \tag{2}$$

Where: $\overline{x_j}$ and S_j, respectively, are the mean and standard deviation of the jth target sample, and

$$\overline{x_j}=\frac{1}{n}\sum_{i=1}^{n}x_{ij} \tag{3}$$

$$S_j=\left[\frac{1}{n-1}\sum_{i=1}^{n}\left(x_{ij}-\overline{x_j}\right)^2\right]^{1/2} \tag{4}$$

(2) Calculation of correlation between the matrix

Based on the standardized data matrix $X^{*}=(x_{ij}^{*})$, calculated the correlation coefficient matrix $R=(r_{ij})m\times m$. Where, r_{ij} are the correlation coefficient between the x_i and x_j target factor .

$$r_{ij}=\frac{1}{n-1}\sum_{k=1}^{n}x_{ki}^{*}x_{kj}^{*}=\frac{\sum_{k=1}^{n}\left(x_{ki}-\overline{x_i}\right)\left(x_{kj}-\overline{x_j}\right)}{\sqrt{\sum_{k=1}^{n}\left(x_{ki}-\overline{x_i}\right)^2\left(x_{kj}-\overline{x_j}\right)^2}} \tag{5}$$

Where, i, j=1,2,…,m.

(3) Solving eigenvalue of the correlation matrix and eigenvectors

Calculating the characteristic equation $\left|R-\lambda I\right|=0$, obtained all of the eigenvalue $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n$, and the

corresponding Tikhonov unit eigenvector t_j=（t_{1j}，t_{2j}，…，t_{mj}）

$$Y_j = \sum_{k=1}^{m} t_{kj} \bullet x_k^* \tag{6}$$

Where: x_k* is the standardized sample matrix.

(3)To determine the number of principal components

Selecting *r* principal components in the m principal components that have been identified to finally realize the evaluation analysis. In general, the contribution rate of variance $e_j = \lambda_j / \sum_{k=1}^{m} \lambda_k$ could explain that principal component Yj reflects the amount of information size. R is determined by the principle that accumulated contribution value $G(r) = \sum_{k=1}^{r} e_k$ is large enough (typically more than 85%). *K* is kth measured values of the ith and jth factor , *k*=1,2,…,*r*.

B. Introduction to Genetic Algorithm

Gas a search technique that imitates the natural selection and biological evolutionary process were first established on a sound theoretical basis by Holland [6-8]. Genetic algorithm has a wide range of, particularly in combinatorial optimization problems and they were proved to be able to provide near optimal solutions in reasonable time[9], it can deal with arbitrary forms of the objective function and constraints, whether it is linear or non-linear, continuous or discrete, in theory, have access to the optimal solution. However, in practical applications of genetic algorithm to demonstrate the more serious question is "premature convergence" problem, less capable local optimization, the late slow convergence and can not guarantee convergence to global optimal solution and so on. In recent years, many scholars try to improve genetic algorithms, such as improving the encoding scheme, fitness function, genetic operator design. However, these improvements are all make in internal of the genetic algorithm and it has been proved that it is unable to overcome these shortcoming effectively.

The most common type of genetic algorithm works like this: a population is created with a group of individuals created randomly. The individuals in the population are then evaluated. The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task. Two individuals are then selected based on their fitness, the higher the fitness, the higher the chance of being selected. These individuals then "reproduce" to create one or more offspring, after which the offspring are mutated randomly. This continues until a suitable solution has been found or a certain number of generations have passed, depending on the needs of the programmer[7].

C. Support Vector Machine

1) Linear Support Vector Machine: The basic concept of SVM regression is to map nonlinearly the original data x into a high-dimensional feature space, and to solve a linear regression problem in this feature space[5-7].

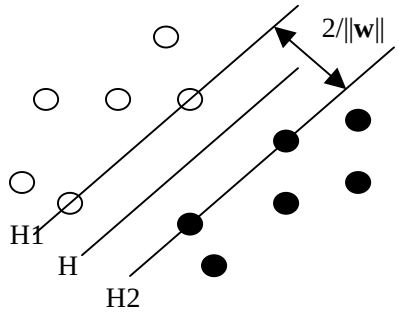


Figure 1. The basic concept of SVM regression

Sample set is established: $\{(x_i, y_i), i = 1,2,\cdots l\}$, Regression function with the following linear equation to represent

$$f(x) = \omega \cdot \phi(x) + b \tag{1}$$

Assumes that all the training data without error ε in the linear function fitting,

$$\min \frac{1}{2}\|\omega\|^2$$

That is

$$st\begin{cases} \omega \cdot x_i + b - y_i \le \varepsilon \\ y_i - \omega \cdot x_i - b \le \varepsilon \end{cases}, i = 1,2,\cdots l \tag{2}$$

Taking into account the permissible error of the case, the introduction of relaxation factor ξ_i, ξ_i^* , then the above formula will change to become

$$\min . \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*)$$

$$s.t. \begin{cases} b - y_i + \omega \cdot x_i \le \varepsilon + \xi_i \\ -\omega \cdot x_i - b + y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, i = 1,2,\cdots, l \end{cases} \tag{3}$$

Where the constant C＞0，expressed the degree of punishment of the sample beyond the error of ε t, Upper and lower limits for the slack variable were ξ_i, ξ_i^* .

Therefore, Lagrange function is constructed:

$$L(\omega, b, a) = \frac{1}{2}\langle \omega \cdot \omega \rangle + C \sum_{i=1}^{l} \left(\xi^* + \xi_i^* \right)$$

$$-\sum_{i=1}^{l} a_i \left[\varepsilon + \xi_i^* + y_i - (\omega \cdot x_i + b) \right] \tag{4}$$

$$-\sum_{i=1}^{l} a_i^* \left[\varepsilon + \xi^* - y_{i+} (\omega \cdot x_i + b) \right]$$

Its dual problem may be

$$\max_{\alpha,\alpha^*}.[\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(x_i\cdot x_j) -\sum_{i=1}^{l}\varepsilon(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)] \quad (5)$$

$$s.t.\begin{cases}\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0\\ \alpha_i,\alpha_i^*\in[0,C/l]\end{cases}$$

Regression function may be:

$$f(x)=\omega\cdot x+b=\sum_{i=1}^{l}(a_i-a_i^*)(x_i,x_j)+b \quad (6)$$

2) Nonlinear support vector machine regression: The problem discussed above is linear, the nonlinear problem, the input sample x by $\psi:\ x\to h$ mapping to high dimensional feature space H (may be infinite-dimensional). When the feature space, construct the optimal hyperplane, the fact simply performs internal product operation, and this inner product operation is in their original space in the function Lai implementation of, not Biyaozhidao the form. As long as the kernel function $K(x_i,x_j)$ satisfies the condition, it corresponds to a transformation that is $K(x_i,x_j)=(\Psi(x_i)\cdot\Psi(x_j))$. Known by the linear support vector regression, quadratic programming Lagrangian objective function:

$$L(\omega,b,a)=\frac{1}{2}\langle\omega\cdot\omega\rangle+C\sum_{i=1}^{l}(\xi^*+\xi_i^*) -\sum_{i=1}^{l}a_i\left[\varepsilon+\xi_i^*+y_i-(\omega\cdot\Psi(x_i)+b)\right] \quad (7)$$
$$-\sum_{i=1}^{l}a_i^*\left[\varepsilon+\xi^*-y_i+(\omega\cdot\Psi(x_i)+b)\right]$$

$$L(\omega,\xi,b,a,\beta)=\sum_{i=1}^{l}a_i-\frac{1}{2}\sum_{j=1}^{l}y_iy_ja_ia_jK(x_{i,}x_j)$$

Dual form:

$$\max_{\alpha,\alpha^*}[\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_j^*)K(x_i\cdot x_j) -\sum_{i=1}^{l}\varepsilon(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)] \quad (8)$$

$$s.t.\begin{cases}\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0\\ \alpha_i,\alpha_i^*\in[0,C/l]\end{cases}$$

Regression function can be:

$$f(x)=\omega\cdot x+b=\sum_{i=1}^{l}(a_i-a_i^*)K(x_i,x_j)+b \quad (9)$$

3) Kernel Functions: In order to get better performances for the support vector machine, an improved method is to combine a number of kernel functions to form a mixed kernel. Mixed kernel function of the form as:

$$K(x_i\cdot x_j)=K_1(x_i\cdot x_j)+K_2(x_i\cdot x_j)$$
$$K(x_i\cdot x_j)=aK_1(x_i\cdot x_j)$$
$$K(x_i\cdot x_j)=K_1(x_i\cdot x_j)K_2(x_i\cdot x_j)$$

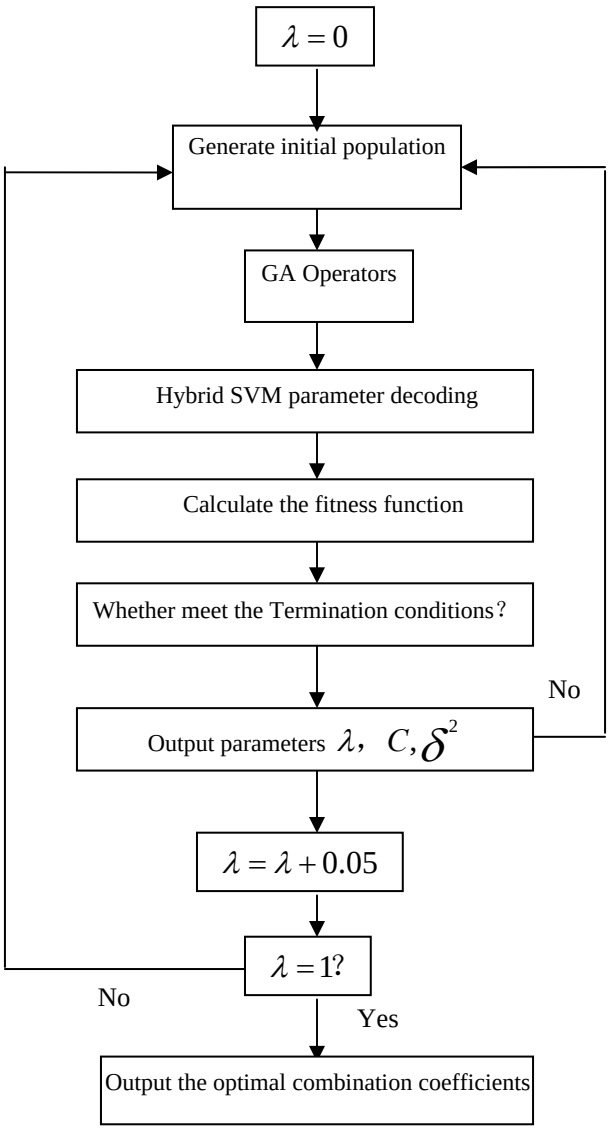


Figure 2. Specific processes of SVM with GA

This paper adopted experimental method to select the combination of the kernel function. In other words, each individual kernel function support vector machine was

established to determine the optimal parameters. Then support vector machine based on several samples of the prediction accuracy to select the best kernel function of two properties, using the combination of the selected two core functions, convex the two combinations to be mixed functions:

$$K(x_i \cdot x_j) = \lambda K_1(x_i \cdot x_j) + (1-\lambda) K_2(x_i \cdot x_j) \quad (10)$$

Which λ is a combination coefficient of adjustment for the two functions mixed, when you select the appropriate parameters, the mixed kernel has both a good learning ability and good generalization ability. When identified two nuclear function and parameters, and then need to determine the kernel function is a combination of two factors, namely, to determine which kernel function in this mixed kernel plays a leading role is our task to be accomplished the following. This paper used genetic algorithm to optimize combination of factors and determine the kernel function according to combination of factors. Specific processes were as shown in Fig.1.

III. APPLICATION CASE

A. General Contracting Risk Factors

General contracting risk is described by 17 indicators are including in such as the War and civil strife I_1, Nationalization I_2, Non-payment of debt I_3. Exchange rate I_4, Prices risk I_5, Pay delay I_6, Design I_7, Construction I_8, Project estimated I_9. Sub-contractors I_{10}, Vendor I_{11}, Geological and foundation conditions I_{12} Hydro-climatic conditions I_{13}. Contract management I_{14}, Financial management I_{15}, Engineering management I_{16} and Public relations I_{17}.

B. General Contracting Risk Analysis Based on PCA

First, the original data in Table1 was processed for the standardization, and by 5, the correlation coefficient matrix was calculated. Then, by the correlation coefficient matrix eigenvalue calculation, as well as all the main components of the contribution rate and the cumulative contribution rate as shown in Table 2. From Table 2, we can see that the first, second and third principal component of the cumulative contribution rate were up to 87.6%, so just find the first, second and the third principal component.

The principal components analysis and fuzzy method were introduced in evaluate the general contracting risk. From Table2, principal components analysis of general contracting risk can be concluded as follow. In the first principal component, War and civil strife I_1, Nationalization I_2 and Non-payment of debt I_3. The first principal component can therefore be considered as Political risk factors. At the second principal components, Exchange rate I_4, Prices risk I_5, and Pay delay I_6, thus, the second principal component can therefore be considered as economic -related risk factors The third principal component of the values of Design I_7, Construction I_8, Project estimated I_9. Sub-contractors I_{10} and Vendor I_{11} were higher, it reflects the technical indicate a high degree risk in the construction. The fouth principal component that including Geological and foundation conditions I_{12} and Hydro-climatic conditions I_{13} can therefore be considered as Natural -related factors. The final principal component that including Contract management I_{14}, Financial management I_{15}, Engineering management I_{16} and Public relations I_{17} can therefore be considered as management -related risk factors.

TABLE I. GENERAL CONTRACTING RISK DATA

No.	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}
1	75	66	1	2	82	0.6	74	74	1	1	2.0	45	0.15	0.14	0.32	88	1
2	80	63	1	1	68	0.6	63	66	1	1	2.4	50	0.20	0.19	0.42	75	1
3	92	98	1	1	79	0.7	70	63	1	1	1.4	38	0.09	0.08	0.20	65	1
4	80	72	1	1	86	0.5	69	98	1	0	0.9	28	0.12	0.21	0.36	83	2
5	90	90	1	2	62	0.5	71	72	1	1	1.8	36	0.14	0.15	0.34	76	2
6	84	69	3	1	84	0.6	72	90	3	1	1.3	32	0.15	0.09	0.35	98	2
7	62	79	0	1	78	0.4	61	69	0	0	1.4	35	0.15	0.09	0.38	70	2
8	94	84	1	2	79	0.7	82	79	1	1	0.6	24	0.16	0.16	0.40	94	1
9	65	86	0	0	61	0.9	88	84	0	0	0.5	20	0.12	0.11	0.38	84	1
10	75	71	1	2	85	0.6	83	86	1	1	0.3	18	0.16	0.16	0.22	65	2
11	66	79	3	0	75	0.6	71	71	3	3	1.8	42	0.09	0.05	0.30	62	1
12	82	80	3	1	65	0.5	80	79	3	1	0.0	12	0.00	0.00	0.38	72	2
13	64	83	0	0	76	0.6	60	80	0	1	1.4	38	0.15	0.14	0.25	63	1
14	94	81	3	1	88	0.7	79	83	3	2	0.7	26	0.12	0.12	0.26	85	2
15	91	85	3	0	73	0.6	70	81	3	1	1.6	34	0.08	0.08	0.34	65	1
16	71	72	2	2	75	0.7	88	85	2	2	0.5	20	0.09	0.15	0.30	90	2
17	88	85	2	2	65	0.5	65	72	2	2	1.4	34	0.00	0.14	0.35	72	2
18	89	80	1	1	76	0.6	88	85	1	1	0.3	12	0.15	0.04	0.22	68	1
19	77	88	3	1	65	0.5	70	81	3	1	1.6	34	0.08	0.08	0.38	72	1

TABLE II. PRINCIPAL COMPONENT ANALYSIS

Indicts	Y1	Y2	Y3	Y4	Y5
I_1	-0.20	-0.19	0.629	-0.450	0.757
I_2	-0.28	0.52	0.436	-0.638	0.214
I_3	-0.44	0.03	0.455	-0.744	0.046
I_4	0.33	0.76	0.333	-0.593	0.065
I_5	0.23	-0.72	0.713	0.590	0.319
I_6	0.47	0.49	0.788	-0.158	0.184
I_7	0.11	0.12	0.130	0.757	-0.168
I_8	0.79	0.93	0.229	-0.121	0.713
I_9	0.73	0.33	-0.321	-0.273	0.788
I_{10}	0.89	-0.251	0.409	0.023	0.130
I_{11}	-0.10	-0.016	-0.110	0.189	0.229
I_{12}	0.78	0.757	0.308	0.096	-0.321
I_{13} .	-0.16	-0.42	-0.216	-0.041	0.409
I_{14}	0.72	-0.23	0.672	-0.130	-0.110
I_{15}	0.812	0.308	-0.184	-0.225	0.184
I_{16}	0.903	-0.216	-0.042	0.125	-0.168
I_{17}	0.86	0.530	-0.016	-0.016	-0.184
Eigenvalue	7.52	2.97	0.757	1.145	0.742
Contribution rate(%)	46. 73	22. 64	8.812	6.354	6.235
Cumulative contribution rate(%)	48.89	52.26	63.56	76.01	83.23

C. *General Contracting Risk Forecasting Based on GA-SVM*

A total of 500 input-output data pairs were obtained for the training of the SVM for real estate prices. Due to the low dimensionality of the parameters pace and the limited range of variation in the parameters, such number of data reasonably covers the set of different possible operating points. The available data set is randomly partitioned into a training set and a checking set[8].

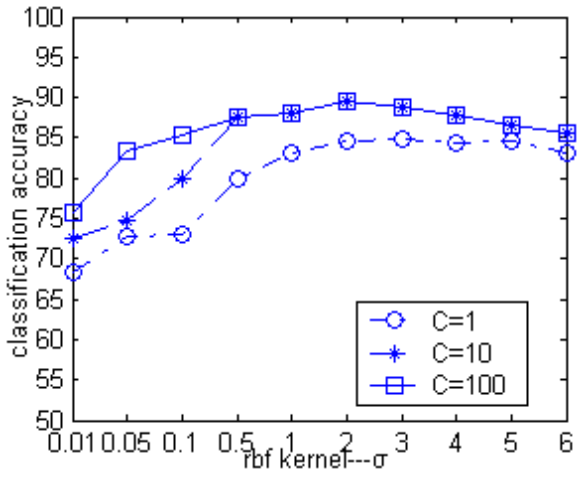


Figure 3. The relationship between classification accuracy and the parameter combination

TABLE III. THE ERROR RATE OF RBF KERNEL FUNCTION ON 2-DIMENSIONAL PROBLEMS OF CATEGORY 2 CLASSIFICATION

σ	C	Error rate
5	0	26%
5	100	49%
0.2	0	15%
0.2	100	20%

TABLE IV. THE ERROR RATES OF DIFFERENT ALGORITHMS

Category Type	Parameter	Accuracy
Polynomial SVM	d=4	86.7%
RBF-SVM	σ =2,C=10(C=100)	88.6%
Mixed kernel function SVM	σ =0.2, C=10	93.3%
RBF-ANN	σ =0.01,C=100	83.3%

IV. CONCLUTIONS

In this paper, SVMG is applied to determination real estate price. The real data sets are used to investigate its feasibility in forecasting the risk of general contracting. SVMG implements the principle of structural risk minimization in place of experiential risk minimization, which makes it have excellent generalization ability in the situation of small sample. And it can change a non-linear learning problem into a linear learning problem in order to reduce the algorithm complexity by using the kernel function idea. In addition, GA can be used to select suitable parameters to determination the risk of general contracting, which avoids over-fitting or under-fitting of the SVM model occurring because of the improper determining of these parameters. The experimental results reveal the potential of the proposed approach for forecasting the risk of general contracting.

It is generally acknowledged that the risk of general contracting was highly complicated and was interrelated with a multitude of factors. It will be advantageous if the parties to a dispute have some insights to some degree. Principal component analysis (PCA) was introduced to analyze the risk of general contracting. principal components analysis of general contracting risk can be concluded as follow. In the first principal component, War and civil strife I_1, Nationalization I_2 and Non-payment of debt I_3. The first principal component can therefore be considered as Political risk factors. At the second principal components, Exchange rate I_4, Prices risk I_5, and Pay delay I_6, thus, the second principal component can therefore be considered as economic -related risk factors The third principal component of the values of Design I_7, Construction I_8, Project estimated I_9. Sub-contractors I_{10} and Vendor I_{11} were higher, it reflects the technical indicate a high degree risk in the construction. The fouth principal component that including Geological and foundation conditions I_{12} and Hydro-climatic conditions I_{13} can therefore be considered as Natural -related factors. The final principal component that including Contract management I_{14} , Financial

management I_{15}, Engineering management I_{16} and Public relations I_{17} can therefore be considered as management - related risk factors.

This paper introduces an hybrid genetic algorithm (HGA) approach to instance selection in SVM for the risk of general contracting. From the above discussion, the following conclusion can be made:

(1)The factors affecting the housing price were quantified with fuzzy sets and reduced by PCA to the inputs of SVM.

(2) The genetic algorithm was adopted to optimize the weights of SVM. The established SVMG model is capable of accurate determinants for housing price with less time and better convergence.

(3) In Simulation tests, the relative error of Mixed kernel function SVMG models smaller than the Polynomial SVM, the RBF-SVM and of RBF-ANN. Thus, the proposed Mixed kernel function SVMG model is capable of more accurate prediction on risk.

REFERENCES

[1] Hayes, R .W. Rerry, J. G. Thompson, P.A. and willmr. G. Risk management in Engineering Construction Implications for Project Managements: London, Thomas Telford Ltd, 1986:221-229.

[2] Dongping Fang, Mingen Li. Risks in Chinese Construction Market-Contractors' Perspective [J]. Journal of Construction Engineering and Management, 2004(11/12):853-861

[3] Temy Lyons, Martin Skitmore.Froject risk management in the Queensland engineering construction industry a survey[J]. International Journal of Project Management 22(2004)51-61

[4] Wenjuan Liu,Qiang Liu ,FengRuan ,Zhiyong Liang ,Hongyang Qiu. Method for Housing Price Forecasting based on TEI@I Methodology. Systems Engineering-Theory&Practice Volume27,Issue7,July2007.

[5] Keethi S.S,Lin C.J; Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel[J] [M];Neural Computation

[6] Holland J.H., Adaptation in natural and artificial system, Ann Arbor, The University of Michigan Press, 1975.

[7] Hollstien, R. B., Artificial Genetic Adaptation in Computer Control Systems, Ph.D. Thesis, University of Michigan, Ann Arbor, MI., 1971.

[8] Booker, L. B., Improving Search in Genetic Algorithms, pp. 61-73, Genetic Algorithms and Simulated Annealing (L. Davis, editor), Pitman, London, 1987.

[9] Jesus Fraile-Ardanuy, P.J.Zufiria. Design and comparison of adaptive power system stabilizers based on neural fuzzy networks and genetic algorithms. Neuro computing 70 (2007) 2902-2912..

[10] K.M.Saridakis , A.J.DentsorasJ. Integration of fuzzy logic, genetic algorithms and neural networks in collaborative parametric design. Advanced Engineering Informatics 20 (2006) 379-399

[11] Chau, K. W., Ng, F F. & Hung, E.C.T.. Developer's good will assignificant in fluence on apartment unit prices[J]. Appraisal J ournal. 2001b,vol69,pp.26-34.

[12] Wenjuan Liu,Qiang Liu ,FengRuan ,Zhiyong Liang ,Hongyang Qiu. Method for Housing Price Forecasting based on TEI@I Methodology. Systems Engineering-Theory&Practice Volume27,Issue7,July2007.

[13] Keethi S.S,Lin C.J; Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel[J] [M];Neural Computation

[14] Cristinini N. and Taylor J.S.: An introduction to support vector machine and other kernel-based learning methods, Cambridge, Cambridge University Press, 2000.

[15] SmitsG F, Jordan E M.Improved SVM Regressionusing Mixtures of Ker-nels. Proeeedings of the 2002 International Joint ConfereneeonNeuralNet-works.Hawaii:IEEE, 2002.2785 一 2790.

[16] Sheng-wei Fei *, Ming-Jun Wang, Yu-bin Miao, Jun Tu, Cheng-liang Liu. Particle swarm optimization-based support vector machine for forecasting dissolved gases content in power transformer oil. Energy Conversion and Management 50 (2009) 1604−160.

[17] Cristinini N. and Taylor J.S.: An introduction to support vector machine and other kernel-based learning methods, Cambridge, Cambridge University Press, 2000.

[18] SmitsG F, Jordan E M.Improved SVM Regressionusing Mixtures of Ker-nels. Proeeedings of the 2002 International Joint ConfereneeonNeuralNet-works.Hawaii:IEEE, 2002.2785 一 2790.

[19] Sheng-wei Fei *, Ming-Jun Wang, Yu-bin Miao, Jun Tu, Cheng-liang Liu. Particle swarm optimization-based support vector machine for forecasting dissolved gases content in power transformer oil. Energy Conversion and Management 50 (2009) 1604−160.

Call for Papers and Special Issues

Aims and Scope.

Journal of Computers (JCP, ISSN 1796-203X) is a scholarly peer-reviewed international scientific journal published monthly for researchers, developers, technical managers, and educators in the computer field. It provide a high profile, leading edge forum for academic researchers, industrial professionals, engineers, consultants, managers, educators and policy makers working in the field to contribute and disseminate innovative new work on all the areas of computers.

JCP invites original, previously unpublished, research, survey and tutorial papers, plus case studies and short research notes, on both applied and theoretical aspects of computers. These areas include, but are not limited to, the following:

- Computer Organizations and Architectures
- Operating Systems, Software Systems, and Communication Protocols
- Real-time Systems, Embedded Systems, and Distributed Systems
- Digital Devices, Computer Components, and Interconnection Networks
- Specification, Design, Prototyping, and Testing Methods and Tools
- Artificial Intelligence, Algorithms, Computational Science
- Performance, Fault Tolerance, Reliability, Security, and Testability
- Case Studies and Experimental and Theoretical Evaluations
- New and Important Applications and Trends

Special Issue Guidelines

Special issues feature specifically aimed and targeted topics of interest contributed by authors responding to a particular Call for Papers or by invitation, edited by guest editor(s). We encourage you to submit proposals for creating special issues in areas that are of interest to the Journal. Preference will be given to proposals that cover some unique aspect of the technology and ones that include subjects that are timely and useful to the readers of the Journal. A Special Issue is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

The following information should be included as part of the proposal:

- Proposed title for the Special Issue
- Description of the topic area to be focused upon and justification
- Review process for the selection and rejection of papers.
- Name, contact, position, affiliation, and biography of the Guest Editor(s)
- List of potential reviewers
- Potential authors to the issue
- Tentative time-table for the call for papers and reviews

If a proposal is accepted, the guest editor will be responsible for:

- Preparing the “Call for Papers” to be included on the Journal’s Web site.
- Distribution of the Call for Papers broadly to various mailing lists and sites.
- Getting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Instructions for Authors.
- Providing us the completed and approved final versions of the papers formatted in the Journal’s style, together with all authors’ contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

Special Issue for a Conference/Workshop

A special issue for a Conference/Workshop is usually released in association with the committee members of the Conference/Workshop like general chairs and/or program chairs who are appointed as the Guest Editors of the Special Issue. Special Issue for a Conference/Workshop is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

Guest Editors are involved in the following steps in guest-editing a Special Issue based on a Conference/Workshop:

- Selecting a Title for the Special Issue, e.g. “Special Issue: Selected Best Papers of XYZ Conference”.
- Sending us a formal “Letter of Intent” for the Special Issue.
- Creating a “Call for Papers” for the Special Issue, posting it on the conference web site, and publicizing it to the conference attendees. Information about the Journal and Academy Publisher can be included in the Call for Papers.
- Establishing criteria for paper selection/rejections. The papers can be nominated based on multiple criteria, e.g. rank in review process plus the evaluation from the Session Chairs and the feedback from the Conference attendees.
- Selecting and inviting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Author Instructions. Usually, the Proceedings manuscripts should be expanded and enhanced.
- Providing us the completed and approved final versions of the papers formatted in the Journal’s style, together with all authors’ contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

More information is available on the web site at http://www.academypublisher.com/jcp/.

(Contents Continued from Back Cover)