# Application of Particle Swarm Optimization Algorithm based on Classification Strategies to Grid Task Scheduling

Shaobo ZHONG*
College of Elementary Education, Chongqing Normal University, Chongqing 400700, China
Email: zshaob@163.com

Zhongshi HE
College of Computer Science, Chongqing University, Chongqing, 400044, China
Email: zshe@cqu.edu.cn

*Abstract—* **Grid task scheduling is a NP-hard problem. In this paper, an optimization algorithm of grid task scheduling is brought forward by using classification strategies to improve particle swarm algorithm. The particle swarm is divided into accurate subgroups for local slow search, commonness subgroups for the cloning strategy processing and inferior subgroups for changing into accurate subgroups to operate the positive and reverse clouds. The experimental results show that the scheduling algorithm effectively achieves the load balancing of resources and preferably avoids falling into local optimal solution and the selection pressure of genetic algorithm and elementary particle swarm algorithm. This algorithm has the high accuracy and convergence speed and so on.**

*Index Terms—***grid computing, task scheduling, cloud model, immune clonal algorithm, particle swarm optimization algorithm.**

## I. INTRODUCTION

Grid computing is a hot spot in current internet research and is the development direction of parallel and distributed processing technology. In grid computing each resource could further schedule the allocated tasks according to own scheduling strategy [1]. Task scheduling is the core of grid computing and also a NP-complete problem. Now the task scheduling becomes the focus in current research field of computing grid. Document [2] proposed a task scheduling of computing grid based on Agent and realized three protocol prototypes of grid resource management. Document [3] proposed an advanced Petri model of time which can be used in the task scheduling of computing grid. Document [4] designed an ant algorithm (AA) of task scheduling of computing grid. Document [5] introduced a grid task scheduling algorithm based on genetic algorithm (GA), which aims at improving the utilization of resource and throughput at utmost. Document [6-7] proposed the application of the particle swam in grid task scheduling, which has a further research in the finishing time of task and the improvement in global searching ability.

Due to the heterogeneity, distribution and extension of grid it is possible that no task could be submitted so that it caused the heavy load of some resource and idle resource is generated when lots of asking tasks are in the grid environment. And this problem is more critical when it involves the scheduling in the cross field and large-scale application. In this paper we propose a particle swarm algorithm based on classified strategy—CSPSO, which could efficiently improve the running efficiency, searching ability of global searching and solution accuracy and guarantee the load balance of grid task scheduling.

## II. THE OPTIMIZED MODEL OF GRID RESOURCE

*Define 1 Grid resource model:* Assumed that the triple $G = (R, T, L)$ represents the grid environment. In which $R = \{r_1, r_i, ..., r_m\}$ represents the congregation of m computing resource and $T = \{t_1, t_2, ..., t_n\}$ means n independent tasks and $L = \{l_1, l_2, ...l_m\}$ means the dynamic load weights of m resource nodes.

Suppose that the CPU utilization, memory utilization, flow rate of current network, I/O access rate of disk and total process number of resource node $r_j$ are represented by $C_j\%$ , $M_j\%$ , $N_j$ , $Io_j\%$ , $P_j$ , so the dynamic load weight of resource node $r_j$ could be represented by:

Manuscript received July 1, 2010; revised January 1,2011;accepted January 22,2011.
This work was funded by the Key Project of Chinese Ministry of Science and Technology (No. 2008ZX07315-001), Major scientific and technological special project of Chongqing (No.2008AB5038)

*Corresponding author: Shaobo Zhong.


doi:10.4304/jsw.7.1.118-124

$$l_j=\pi_1\times C_j\%+\pi_2\times M_j\%+\pi_3\times N_j+\pi_4\times Io_j\%+\pi_5\times P_j \quad (1)$$

In equation (1), $\sum_{i=1}^{5}\pi_i=1,\pi_i$ means the importance level of each load parameter.

For m nodes, the three resources of each node are running time, transmission delay of network and resource throughput, respectively. We name them as $(R_{p_j})_i,(i=1,2,...,m;j=1,2,3)$. In which the source requirements of each task to each node are $(R_{q_j})_i,(i=1,2,...,m;j=1,2,3)$. The unit costs of the three types of resources are c1, c2 and c3. And the submitted task number M is larger than the node number of grid R. The cost weights are w1, w2 and w3. So the total cost of ith node where the ith task is allocated is

$$O_{ij}=\sum w_k c_{ik} R_{qk} \quad (2)$$

In which $i=1,2,...,M;j=1,2,...m;c_{ij}$ is the cost matrix of three types of resources for each task.

The upper limit of cost of task finishing, named (Cost)i(i=1,2,⋯,M), that is the satisfaction degree of user, is submitted when the task is submitted.

Based on the analysis above and better solving the problem of allocating multiple tasks in one node, the min-max target function and relevant constraint conditions could be built.

Min-Max target function is

$$\min\{f(T)\}=\sum_{i=1}^{M}\left(\sum_{j=1}^{m}(Q_{ij}N_{ij})+\sum_{k=1}^{i-1}(Q_{ik}N_{ik})\right) \quad (3)$$

$$\max\{E(T)\}=\sum_{i=1}^{M}\frac{Q_{ij}}{\sum_{j=1}^{m}(Q_{ij}N_{ij})+\sum_{k=1}^{i-1}(Q_{ik}N_{ik})} \quad (4)$$

The equation (3) is the optimal scheme of grid resource allocation and task scheduling, which realizes the minimum cost of resource utilization. And this scheme is realized by particle swarm algorithm and cloud model algorithm. And f(T) means the total cost of task running.

The equation (4) makes the utilization of task scheduling running to the most in comparison with the minimum cost of equation (3). E(T) means the total efficiency of task running.

## III. THE TASK SCHEDULING ALGORITHM BASED CSPSO

### A. Basic particle swarm algorithm

PSO algorithm, which is an algorithm of intelligent optimization of swarm, was proposed by Kennedy and Eberhart in 1990s [8]. Its idea derives from the research of preying behavior of birds. The PSO algorithm provides a kind of advantage for the evolution of swarm based on the sharing mechanism of society information in biology groups. In PSO algorithm, the solution of each optimized problem is a particle of searching space and each particle has own position and velocity. All the particles have the adaptation value decided by an optimized function and look for the optimal solution by the cooperation of particles in solution space. In comparison with the genetic algorithm and ant colony algorithm, PSO is simple, easily realized and has less adjustment parameters. Currently this algorithm is successfully applied to the structure design, neurotic network, electromagnetic, plan of multiple tasks, task scheduling and other optimized engineering problems.

Definition 2 the velocity and position of the ith particle in n-dimension searching space in time t+1 are defined as

$$v_i(t+1)=\omega v_i(t)+c_1r_1(p_i-x_i(t))+c_2r_2(p_g-x_i(t)) \quad (5)$$

$$x_i(t+1)=x_i(t)+v_i(t+1) \quad (6)$$

In equation (5) and (6), $t$ is the evolving algebra. $v_i(t)$ is the velocity of iteration t of particle i. $v_i\in[-v_{i\max},v_{i\max}],x_i(t)$ is the position of iteration t of particle i. $\omega$ is the inertia weighing coefficient. Learning factor c1 and c2 is non-negative constant and generally cl=c2=2. $r_1,r_2\in[0,1]$ is the random number which follows an even distribution. In which the current position of particle i in n-dimension solution space is represented by $x_i=(x_{i1},x_{i2},...,x_{in})$ and the current velocity is represented by $v_i=(v_{i1},v_{i2},...,v_{in})$. $p_i$ means that the particle i itself finds the optimal solution, called individual extreme. $p_g$ means the optimal solution which could be found in the whole species group and this value is a global extreme. Particle updates its position and velocity according to equation (5) and (6).

Definition 3 The current optimal and individual extreme $p_{i+1}$ of particle i is decided by

$$p_i(t+1)=\begin{cases}p_i(t), if\ f(x_i(t+1))\ge f(p_i(t))\\ x_i(t+1), otherwise\end{cases} \quad (7)$$

*Definition 4* The adaptation function $F(x)$ of particle x is

$$F(x)=\frac{1}{\min(f(x))} \quad (8)$$

*Definition5* The current global extreme $p_g$ of particle i is decided by

$$p_g(t)\in\{p_0(t),p_1(t),...,p_n(t)\mid f(p_g(t))=\min\{f(p_0(t)),f(p_1(t)),...,f(p_n(t))\}\} \quad (9)$$

### B. *Cloud Model*

*Definition 6 Cloud* Model Assume X is a normal set named domain [9, 10] and X={x}. Regarding the fuzzy set A~ in domain X, any x has a random $\mu A\sim(x)$ named membership of x to A~ and this random number has a steady tendency. The distribution of membership of basic variants is cloud. The features of cloud is characterized by expectation $E_x$, entropy $E_n$ and excess entropy $H_e$.

The normal cloud is the most important and powerful tool which is used to describe the language atom, such as youth, high salary, which could be described well by cloud.

*Definition 7* Expectation $E_x$: In universe X of common normalized cloud, the basic variable x which has a biggest value is called the expectation of cloud, which calibrates the location of cloud object in universe. And it means the position of gravity center, which indicates the center value of information of relative fuzzy concept.

Entropy $E_n$: the metric of fuzzy level. The size of entropy directly decides the range accepted by the fuzzy concept in universe.

Excess Entropy $H_e$: The so-called entropy of entropy $E_n$ and it indicates the discrete level of cloud. The size of Excess Entropy indirectly indicates the thickness of cloud.

As shown in Fig.1, a basic normal cloud could be described fully only by the expectation $E_x$, entropy $E_n$ and excess entropy $H_e$.

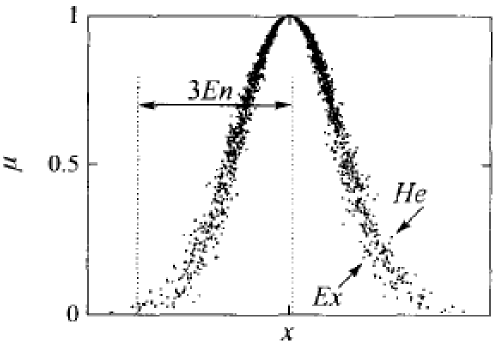
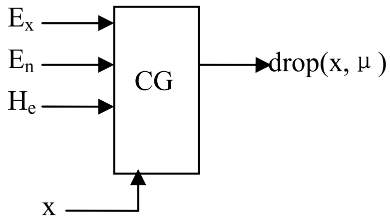
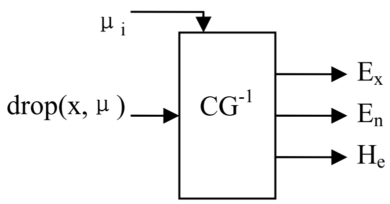



Figure 1. The normal cloud model.

The Generator of Cloud has two types, forward one and backward one, as shown in Figure 2. The forward generator generates the two-dimension cloud $drop(x,\mu)$ named cloud particle which satisfies the rule of normal cloud above based on the three numeric characteristics of current normal cloud. The backward generator is the cloud particle with a considerable number of quantity, $drop(x,\mu)$, which determines the three numeric characteristics $E_x, E_n, H_e$. Regarding the particle swarm in grid resource allocation, the particle in elite particle swarm could be regarded as cloud particle, which generates the relevant numeric characteristics $E_x, E_n, H_e$ by forward cloud and then modifies the value of $E_n$ and $H_e$ and generates the same number of cloud particle by backward cloud algorithm. Thus the elite sub-swarm is generated and the final solution is gradually approached.




(a) Forward generator of cloud




(b) Backward generator of cloud

Figure 2. Generator of cloud.

When the number field corresponding to the concept is one-dimensional, the algorithm of normal cloud generator is as follows:

*Algorithm 1: Forward cloud algorithm*

Input: three digital eigenvalues $E_x, E_n, H_e$ representing the qualitative concept $\widetilde{A}$;the number of cloud droplet N;

Output: quantitative values of n-cloud droplet $\{x_1, x_2, \dots, x_n\}$ as well as the indeterminacy of cloud droplets representing concept $\widetilde{A}$.

Steps:

generate a normal random number $E_n^{'}$ with $E_x$ as expected value and $H_e$ as standard deviation

generate a normal random numberx with $E_x$ as expected value and the absolute value of $E_n^{'}$ as standard deviation

Set x is a specific quantitative characterization of the qualitative concept $\widetilde{A}$,known as cloud droplets;

Compute $y = e^{\frac{-(x-E_x)^2}{2(E_n^{'})^2}}$

Set y is the indeterminacy of x belonging to the qualitative concept $\widetilde{A}\ \{x, y\}$ completely reflects the entire content of the qualitative and quantitative conversion.

Repeat steps 1-6, until n-cloud droplets are generated

*Algorithm 2. Reverse Cloud Algorithm*

Input : n-cloud droplets $\{x_1, x_2, ..., x_n\}$

Output: expectation $E_x$, entropy $E_n$ and excess entropy $H_e$ of the qualitative concept represented by n-cloud droplets

Steps:

according to xi, calculate the sample mean $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$ of this set of data, the absolute central moment $\frac{1}{n}\sum_{i=1}^{n}|x_i - \overline{X}|$ of first-order samples, sample variance $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{X})^2$, the estimate of $E_x$ is $\hat{E}_x = \overline{X}$, the estimate of $H_e$ is $\hat{H}_e = \sqrt{\frac{\pi}{2}} \times \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{E}_x|$, the estimate of $E_n$ is $\hat{E}_n = \sqrt{S^2 - \frac{1}{3}\hat{H}_e^2}$.

### C. Immunity clone algorithm

The immunity clone algorithm simulates the multiple-clone mechanism of biological immunity system, which not only adopts the variation and cross operation to realize the information exchange of antibodies but also utilizes the obtained characters of antigen reaction to increase the clone diversity during the change process of antibodies. The clone operator is based on affinity degree function and splits one point into multiple same points, which experiences clone variance, clone cross and clone selection variance to obtain new antigen swarm [11-15]. During the procedure of clone algorithm running, the strategy of clone copy to father generation is adopted in order to keep the diversity of solution and enlarge the searching range of space. And the enlarging of solution space is at the cost of time increase.

The procedure of immunity clone algorithm is

(1) Initiating. Assume the end conditions of algorithm and give the variance probability $p_m$, clone scale $n_c$, antigen swarm scale n, antigen swam updating ratio T, antigen coding size l. The initial antigen swarm A(0) is generated randomly and s antigens in A(0) are selected to become the memory unit M and the evolution algebra g =0;

(2) Calculate the compatibility $f(x)$;

(3) Judge the end condition of algorithm. End the algorithm if satisfies this condition. Otherwise continue.

(4) Add 1 to evolution algebra g and carry out the generation operation to antigen swam and memory unit.

(5) Antigen swarm updating. The T memory units whose compatibility is stronger and individuals of antigen swarm substitute T antigens whose compatibility is weaker. Then the new antigen swarm $A^{'}(g)$ is generated.

(6) Clone. The swarm $Y(g)$ is obtained by carrying out the clone operation to father swarm of the gth generation.

(7) Variance. The swarm Z(g) is obtained by carrying out the variance operation to swarm $Y(g)$ according to probability $p_m$.

(8) Clone selection operation. The antigen swarm A(g+1) of next generation is obtained in swarm Z(g) by selecting the individuals whose compatibility is stronger based on curtain ratio.

(9) Local searching of memory unit M(g). The local searching is carried out by selecting the individual whose compatibility is the strongest in antigen swarm A(g+1) which did the clone selection. And the individual whose compatibility is weakest is substituted if the maximum value of searching feedback is larger than the minimum compatibility value in memory swarm. And then the memory unit M(g+1) of next generation is obtained by carrying out the local searching to M(g).

### D. The description of algorithm procedure

(1) Init and parameter determination. Suppose that swarm $X(T_i) = \{x_1(T_i), x_2(T_i), ..., x_N(T_i)\}, i = 0,1,...,\max$, in which N is the swarm scale. And learning factor is $c_1, c_2$, the hybrid probability is $p_c$, the variance probability is $p_m$, inertia weighting factor is $\omega_0$ and initial swarm $X(T_0)$ of N particles are generated randomly.

(2) The temporary swarm $X^{'}(T_i)$ is generated by carrying out the operation to current swarm $X(T_i)$ based on formula (5) and (6).

(3) Order the particles in swarm $X^{'}(T_i)$ based on the adaptation value and generate three sub-swarms, $X_H^{'}(T_i)$, $X_M^{'}(T_i)$ and $X_L^{'}(T_i)$, whose particle number is the same based on adaptation values from high to low.

(a) Carry out operations to sub-swarm $X_H^{'}(T_i)$ based on formula (4) and (5) continuously. The value of $\omega$ is very small so the velocity change is very slow. Then the feasible solutions could be searched in smaller local range because the particles in sub-swarm $X_H^{'}(T_i)$ are elite particles and the probability reaching the feasible solution is the largest.

(b) Copy all the particles in sub-swarm $X_H^{'}(T_i)$ to sub-swarm $X_L^{'}(T_i)$ and regard the particle as cloud particle to determine Ex, En and He by backward generator. And by obtained Ex and modified value of En and He random quantity of cloud particles are generated using forward cloud operator. By such operations, the operations such as inheritance, variance and intersect cross is carried out to improve the characters of ill particles.

(c) The sub-swarm $X_M^{'}(T_i)$ could be carried out based on immunity clone algorithm.

(4) If the current optimal individual satisfies the convergence conditions the evolution process is ended successfully and the results are retuned. If the evolution times do no reach the pre-defined maximum evolution times, then the procedure goes to step (2).

## V. SIMULATION EXPERIMENT AND ANALYSIS

### *A. Experiment environment*

This paper did the simulation experiments for the task scheduling algorithm in order to verify the correctness of algorithm and evaluate its performance. In experiment, we simulate the grid task scheduling in grid computing environment which constitutes four station nodes. The resource situation of each node is shown in Table 1.

TABLE 1

SIMULATION ENVIRONMENT

| The name of node | Configuration of clusters | The quantity of clusters | Manageme nt software |
|---|---|---|---|
| site1 | 64 nodes | 8 | LSF 6.0 |
| site 2 | 128 nodes | 4 | LSF 6.0 |
| site 3 | 128 nodes | 2 | LSF 6.0 |
| site 4 | 256 nodes | 2 | LSF 6.0 |

### *B. The performance comparison of load balance*

The main task of experiment is checking the performance of grid load balance of CMPSO algorithm and GA algorithm. The main parameters of CMPSO are:

The size of swarm is 100. The learning factor of self-consciousness equals to $c_1 = 1.46$. The learning factor of swarm-consciousness equals to $c_2 = 1.46$, $\omega_0 = 0.9$. The evolution algebra is 20. The threshold value λ local=2, λ global=10 and evolution coefficient K=10. The results of experiment are shown in Figure 3 and Figure 4.

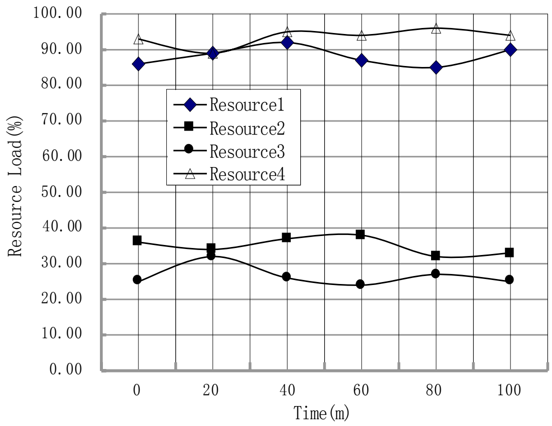



Figure 3. The load statistics of resource before running the CMPSO.

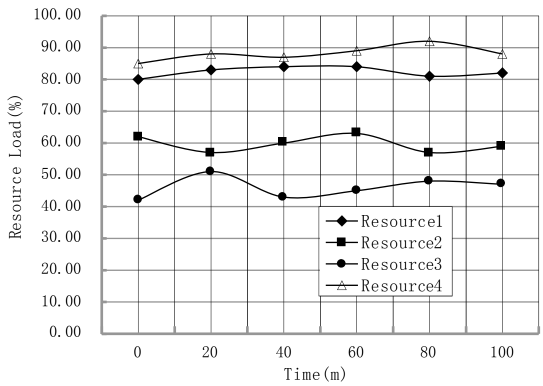



Figure 4. The load statistics of resource after running the CMPSO.

In figure 3, we find that part of grid users could not continue to submit tasks because the resource 1 and resource 4 have too many tasks so that the load of resource is very heavy. But the load of resource 2 and resource 3 is too low so the resource does not get utilized fully and the grid computing could not show good performance. In figure 4, we find that the load of the whole resource is more even with the decrease of resource 1 and resource 4 and the large increase of load of resource 2 and resource 3.

### *C. The comparison among three algorithms*

This test targets to the running efficiency and solution accuracy of CMPSO algorithm, conventional GA algorithm and classic PSO algorithm. The parameters of CMPSO algorithm is the same as that of experiment 1. And 50 generations per time are set. The conventional GA algorithm [16-18] uses the float coding, non-uniformity variation operator and arithmetic cross operator. The classical algorithm adopts the parameter setting of [17]. CMPSO algorithm, GA algorithm and PSO algorithm compare the optimal solution by carrying out 300 experiments for the resource 3.

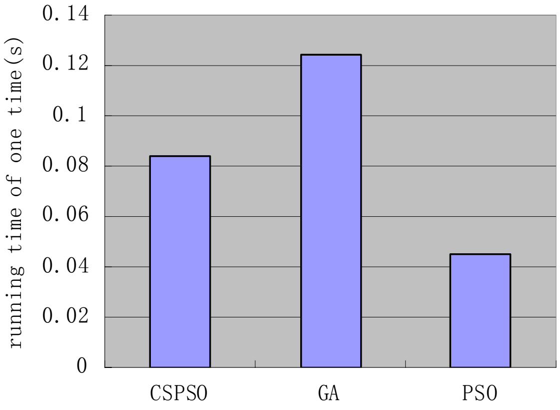



Figure 5. The running efficiency of three algorithms.

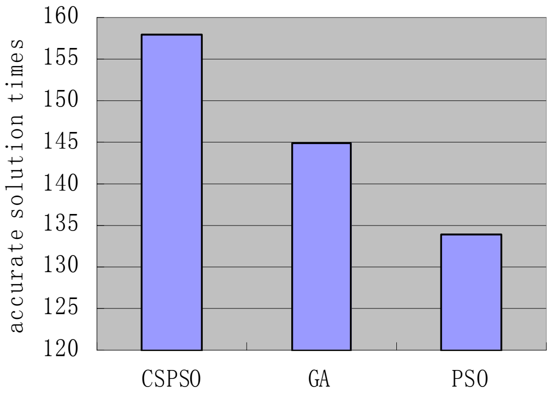



Figure 6. The solution precision of three algorithms.

In figure 5, we could find that the time of one running of CMPSCO is 0.07806s and the running efficiency is higher than that of GA (0.12458s) but lower than that of PSO（0.04532s）.The reason is the CMPSO carries out the classification and immunity clone to the classified common sub-swarms and multiple forward cloud and backward cloud procedure to the inferior sub-swarms during the iterating time. Thus the time efficiency is lower than the PSO algorithm.

But in figure 6, we could find the solution precision is the highest and gets 157 optimal solutions in 300 experiments instead of 145 in GA algorithm and 134 in PSO, which has the lest times of precision solution. This is because CMPSO carried out the procedure of substitution and cloud model to inferior sub-swarms and slow local searching to the elite particles and immunity clone to the common sub-swarms during the running process. So the reliability of solution gets guaranteed.

## VI. Conclusion

The resource management and task scheduling are the two key technologies in grid. The proposed particle swarm algorithm based on classification not only guarantees the optimization allocation of global resource but also considers the load balance of resource.

## Acknowledgment

The authors thank the editor and referees for their careful review and valuable critical comments. We also thank Prof. He for valuable suggestions and comments. This work is supported by the Key Project of Chinese Ministry of Science and Technology (No. 2008ZX07315-001), Major scientific and technological special project of Chongqing (No.2008AB5038), Education project of Chongqing Normal University (080201), the Chongqing Key Research Base of Humanities and Social Sciences: the Financial Support from Chongqing Research Center of Elementary Teacher Education. The authors are grateful for reviewers who made constructive comments.

## References

[1] Rajkumar Buyya, David Abramson, Jonathan Giddy. Grid Resource Management, Scheduling, and Computing Economy. The 1st International Workshop on Grid and Cooperative Computing, Tokyo, Japan: IEEE Computer Society, 2000, pp: 1734-1739.
[2] Li Chunlin, Lu Zhengding, Li Layuan. An Agent-based Approach for Grid Resource Management, Journal of Wuhan University of Technology, 2003, 29(15): 110-115.
[3] Yao jun Han, Resource Scheduling Algorithm for Grid Computing and Its Modeling and Analysis Using Petri Net. The 2nd International Workshop on Grid and Cooperative Computing, Shanghai, IEEE Computer Society, 2003: 814-820.
[4] Zhi hong Xu, Xiangdan Hou. Ant Algorithm-based Task Scheduling in Grid Computing. Proc of Conf on Electrical and Computer Engineering, Quebec Canadian: IEEE Computer Society, 2003: 1107--1110.
[5] Vincenzo Di Martino.Schduling in a grid computing environment using genetic algorithm. The 16th Int’1 Parallel and Distributed Processing Symp, Pittsburgh, USA: Kluwer Academic press, 2002: 678-686.
[6] JI Yi-mu; WANG Ru-chuan. Study on PSO algorithm in solving grid task scheduling, Journal on Communications, 2007, 28(10): 60-66.
[7] YI Kan; WANG Ru-chuan. Nash Equilibrium Based Task Scheduling Algorithm of Multi-schedulers in Grid Computing, Acta Electronica Sinica, 2009, 37(2): 329-333.
[8] Li Deyi, Meng Haljan, Shi Xuemei. Membership Clouds and Membership cloud Generators, Journal of Computer Research and Development, 1995, 32(6): 15-20.
[9] Zhang Guang-Wei, He Rui, Liu Yu, Li De-Yi, Chen Gui-Sheng. An Evolutionary Algorithm Based on Cloud Model. Chinese Journal of Computers, 2008, 31(7): 1082-1091.
[10] Xie Xiao-feng, Zhang Wen-jun,Yang Zhi-lian. Overview of particle swarm optimization. Control and Decision, 2003, 18(2): 129-134.
[11] F. Xie. .Orthogonal Immune Clone Particle Swarm Algorithm on Multiobjective Optimization, Journal of Electronics & Information Technology, 2008, 30(10): 2320-2324.
[12] B Jia. Feature Selection via Immune Memory Clone, Journal of Xi'an Jiaotong University, 2008, 42(06): 0679-0682.
[13] W. Wang. Quantum Clonal Evolutionary Algorithms. Acta Electronica Sinica, 2003, 31(12A): 2066 – 2070.
[14] Y Xi. Quantum-Inspired Immune Clonal Algorithm for SAT Problem. Chinese Journal of Computers, 2007, 30(2): 176-183.

[15] X. Zhang. Development and Prospect of the Artificial Immune System. Acta Electronica Sinica, 2003, 31(10):1540 - 1548.
[16] Liang J J, Qin A K. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computing, 2006, 10(3): 281-295
[17] Zhong Shao-bo. Optimal resource model and task scheduling algorithm based on dynamic load balancing strategy in grid, Journal of Computer Applications, 2008, 28(11): 2867-2870.
[18] Ren Wei-zheng, Zhong Shao-bo, Shu Wan-neng. Resource Matching and Task Scheduling Based on Immune Clonal Strategy in Grid Environment, Journal of PLA University of Science and Technology(Natural Science Edition), 2008, 9(5): 480-482.

**Shaobo Zhong** was born in Sichuan, P.R. China, in January 24, 1973. He obtained the bachelor's the master's degree in Mathematics and Computer Science of the Chongqing Normal University, China in 1998, and the doctor's degree in College of Computer Science of the Chongqing University, China in 2008.His research interest includes machine learning, data mining and web page classification.