International Journal of Multimedia and Ubiquitous Engineering
Vol.12, No.2 (2017), pp.195-204
http://dx.doi.org/10.14257/ijmue.2017.12.2.15

Cloud Computing Scheduling Strategy Based on Multi-Group Parallel Particle Swarm Optimization

Zhang Chun-na[1] and Li Yi-ran[2]

[1]*School of Software, University of Science and Technology Liaoning, Anshan Liaoning 114051, China;*
[2]*College of Applied Technology, University of Science and Technology Liaoning, Anshan Liaoning 114011, China*
E-mail: zcn1979@163.com

Abstract

In order to improve the efficiency of resource scheduling in cloud computing environment and maintain low consumption, the new algorithm based on multi-group parallel Particle Swarm Optimization algorithm is proposed, due to the complexity of the operating environment, the whole population based on the standard Particle Swarm Optimization algorithm is divided equally, the improved algorithm is used to obtain the optimal position of the particle in the subgroup, and then the optimal solution is obtained, for the premature convergence problem of Particle Swarm Optimization algorithm, determining the particle of the poor performance, and using the characteristics of Cauchy distribution to implement perturbation, which can effectively help the algorithm to converge quickly. Experimental data show that the improved algorithm compared with other algorithms, it has achieved good promotion, such as in platform efficiency, consumption and convergence, and so on.

Keywords*: cloud computing; multi-group; Cauchy distribution; Particle Swarm; convergence*

1. Introduction

Cloud computing is the new integrated technology, which integrates distributed computing, grid computing and virtualization technology, it coordinate the large number of distributed resources in the shared network, cut the big data into several sub tasks, and provide the corresponding calculation and service [1-3], therefore, cloud computing resource scheduling problem is a key factor to measure the performance of cloud environment. How to use effective methods to allocate the resources in the cloud environment efficiently and rationally has become the hot research topic.

There are many ways to optimize the resource scheduling of cloud computing [4-5], which can be divided into two categories, static method and dynamic method. The static method can refer to the grid computing, and the method is relatively simple, assuming that the relevant information in the calculation can be measured to achieve the simple distribution of tasks, its deficiency is the resource application and release in the algorithm will greatly improve the system overhead, this will cause the load of resources in the cloud computing environment is not balanced. For the resource scheduling problem of high complexity in cloud computing, it is obvious that more effective method should be adopted [6], based on this, in the face of the problem, it is better to use dynamic intelligent optimization algorithm to solve. Some mainstream algorithm include: genetic algorithm, fish-swarm algorithm, Particle Swarm algorithm, leapfrog algorithm, *etc.* All of the above are simulated the biological behavior in the nature, with adaptive, but still there are many problems, such as poor stability, slow convergence rate, low precision and so on.

ISSN: 1975-0080 IJMUE

The resource scheduling problem in cloud computing is the typical NP problem [7-10]. In this paper, the improved Particle Swarm algorithm of multi-group parallel is used to implement scheduling strategy, it referred to as IMPSO. The algorithm mainly investigates the time cost and energy consumption of the cloud computing scheduling strategy, and establishes the corresponding fitness function model, the whole population is divided into subgroups, the new speed and position formula are used to calculate the fitness of the particle in the subgroup, the local optimum is obtained, and the global optimum is obtained. In order to overcome the defects of the algorithm is easy to fall into local optimum, set particle perturbation mechanism, disturbance the particles of stagnant or oscillating by the characteristic of Cauchy distribution, and the convergence is improved to ensure the smooth running of the following resource scheduling.

2. Cloud Computing Resource Scheduling Strategy

Cloud computing is developed from distributed processing and parallel computing, it is a kind of virtual, on-demand service model. At present, the mainstream programming model uses the Map/Reduce model proposed by Google, which will divided the task into two stages according to the order, Map stage and Reduce stage. Both need to consider the problem of parallelism, the smaller the refinement of Map tasks, the more conducive to parallelism, but this will increase the system load. The Reduce task is required to wait for the slowest Reduce task is completed, the Reduce phase processing is completed. Here, the task division and resource scheduling is reasonable or not will directly affect the efficiency of the task itself.

The definition of virtual resource allocation framework in cloud computing environment is as follows: set the overall task, that is, the total load is T, it is divided into several independent sub tasks, using $T=\{t_1,t_2,\cdots t_m\}$ to express; set R to represent resources, can be expressed as $R=\{r_1,r_2,\cdots r_n\}$, here, the number of resources is m, the number of tasks is n. The mathematical model of the corresponding scheduling strategy is $F=\{T,R,f,P\}$. Among them, f is the objective function, and P is the algorithm. All tasks can be assigned, and a virtual resource only supports a task to run, set the set $E=\{e_1,e_2,\cdots e_n\}$ is a collection of physical devices. Based on this, the cloud computing resource scheduling mathematical model should have the following characteristics:

1) About the resource r_i, it can only face a task;

2) In the platform, applying for resources for each user may have m task, tasks are independent of each other and are performed independently, total task execution time can be expressed as $S_{m\times n}=\{s_{ij}\}$;

3) According to the relationship between t_i and r_i, set t_i on the e_i to perform the consumed time is $TE(t_i,e_i)$, $te_{i,j}$ is the time for the i load to execute on the j physical device, for n tasks of a user, the completion time of all tasks can be expressed as

$$S(t_i)=\sum_{j=1}^{m}\max_{1\le i\le n}(TE(t_i,e_j)).$$

3. Particle Swarm Optimization Algorithm

3.1. Basic Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a bio-simulation heuristic algorithm [11-12], its basic principle is the assumption that there are n particles in the certain space, each particle has the initial position and velocity, the particle follows the trajectory of the optimal particle, and a position represents a solution. The optimal particle position is the optimal solution when the algorithm completes the termination [13-15]. The parameters of this paper are as follows: the position of the i particle in Q dimensional space is $x_i = \{x_{i1}, x_{i2}, \cdots x_{iQ}\}$, the velocity of the particle is $v_i = \{v_{i1}, v_{i2}, \cdots v_{iQ}\}$, $i = 1, 2, \cdots, n$. The global optimal extreme value is Gb, the local optimal extreme value is Pb. The updating of the position in the course of the particle travels with reference to the above two, specific particle state formula is as follows:

$$v_{iq}^{'} = \alpha v_{iq}^{"} + \beta^0 (x_{Pb} - x_{iq}) + \beta^1 (x_{Gb} - x_{iq}) \tag{1}$$

$$x_{iq}^{'} = x_{iq}^{"} + v_{iq}^{'} \tag{2}$$

In the formula, α is the inertia weight, which is used to balance the velocity relation in the flight of the particle. β^0、β^1 is correction factor, $\in [0,1]$, and it is also randomly distributed.

3.2. Chaos Mechanism

Chaos phenomenon is a kind of nonlinear behavior that exists universally in nature, which has the characteristics of ergodicity and regularity, and is sensitive to the initial conditions. It can no-repeated searches within the limited local area, in this way, the characteristic of chaos search can be used to optimize the algorithm, so that the particle can escape from local optimum [16-17]. In the algorithm, if the particle falls into the local optimum, then the new space with the current population in the same dimension is randomly generated, and the chaotic sequence is mapped, to determine whether to replace the current particle by comparing the value of the photographic reading value in the iteration. Specific search steps are as follows:

(1) Define initial area, set N dimensional initial state vector is $R_0 = (R_{01}, R_{02}, \cdots, R_{0N})$, the values in R_0 are adjacent to each other, and the difference is small.

(2) The initial vector R_0 is calculated by using the logistics equation to generate chaotic sequences $c_1, c_2, \cdots, c_n$. Here, after several iterations, the system will be completely in the chaotic state. The vector layer can be expressed as:

$$c_{i+1} = c_i (1 - c_{i-1}) \lambda \tag{3}$$

In the formula, λ is the iterative control parameter.

(3) Setting the space particle is X_i, using the formula (3) to get a better position for X_i, denoted by $X_i^{'}$.

$$X_i^{'} = r \cdot rnd \cdot c_j + X_i \tag{4}$$

In the formula, r is the active radius of particle X_i, $rnd \in [-1,1]$, $j \in [0, n]$.

The main ideas of the Particle Swarm Optimization algorithm based on chaos mechanism are as follows: On the one hand, using chaotic sequence initialize the position and velocity of particle, because of the characteristics of ergodicity, it can not only keep the diversity of particle, but also enhance the searching ability of particle; moreover, the chaotic state can make the motion of the particle have continuity.

Chaos initialization: the initial value of X_i is given in the formula (4), and the velocity of Particle Swarm iteration is modified:

$$\begin{aligned} v'_{i,j}(t+1) = {} & a v''(t) + b^0(t)(x_{lb}(t) - x_{i,j}(t)) \\ & + b^1(t)(x_{gb}(t) - x_{i,j}(t)) \end{aligned} \tag{5}$$

In the formula, a is a constant, $\in (0,1]$, b is a random number of the normal distribution $N[0,1]$, $i \in [1,n]$, $j \in [1,m]$, n is the number of particle, m is the spatial dimension. For $v''(t)$:

$$v''(t) = \begin{cases} v_{i,j}(t) & q = 0 \\ N[0,1] \cdot d \cdot \tilde{v} & q = 1 \end{cases} \tag{6}$$

$$q = \begin{cases} 0 & f(x_{gb}(t-1)) > f(x_{gb}(t)) \\ 1 & f(x_{gb}(t)) = f(x_{gb}(t-1)) = \\ & \cdots = f(x_{gb}(t-5)) \end{cases} \tag{7}$$

In the formula, $\tilde{v} = v_{\max} \cdot c_i / 1.1$, $d = f(x_{gb}) - f(x_T)$, c_i is the new chaotic sequence, $f(x_{gb})$ is a satisfactory solution, $f(x_T)$ is the target solution.

3.3. Particle Distribution

In this paper, the Cauchy distribution is used to adjust the particle state and to improve the diversity of the particle. The idea of Cauchy distribution is that under the premise of getting the fitness value of particle, the disturbance is implemented according to the Cauchy disturbance formula, the new particle position is randomly generated and compared with the original position, if it is superior, then replace with it. The corresponding Cauchy distribution random variable is $C(0,\alpha)$, the density functions are as follows:

$$\phi(x) = \frac{\alpha}{\pi(\alpha^2 + x^2)} \qquad -\infty < x < +\infty \tag{8}$$

In the formula, α is the adjustment parameter, $\phi(x)$ is the standard Cauchy function.

It should be noted that the speed of approaching 0 along the X axis, the Cauchy function is much slower than the probability density function PDF, which means that the Cauchy distribution is more random and the disturbance is more powerful, and can also effectively solve the local optimal problem of Particle Swarm Optimization algorithm.

In the iterative process of the algorithm, for the particle of the local stagnation or repeated shock implement disturbance, and the corresponding position is changed. The formula for triggering the disturbance is as follows:

$$\frac{\sum_{i=k'}^{k'-k''} | f(x_i) - f(x_{i-1}) |}{k''} \le \theta \tag{9}$$

In the formula, k' is the current iteration number and k'' is the previous iteration number of the algorithm, $f(x_i)$ is the fitness value of the particle in the current iteration, θ is the disturbance threshold.

Thus, through comparing the fitness value of the particle, if the difference of the fitness of the multi-generation particle satisfies the disturbance condition, it means that the particle is trapped into the local optimal bound, and the discrete advantage of the Cauchy distribution can effectively increase the diversity of particle, which ensure that the algorithm can obtain the global optimal solution.

3.4. Multi-group Parallel

In order to increase the diversity of particle and make the algorithm converge quickly, this paper proposed the multi-group parallel model, which divides the whole population into several subgroups of equal size, each subgroup is implemented the improved PSO algorithm, and the optimal position of all subgroups is analyzed and compared, The velocity of particle motion is updated by the fusion of the global optimal particle velocity. At the same time, the adjustment factor is introduced into the subgroup to accelerate the convergence of subgroup, the improved velocity update formula is as follows:

$$v_{ik}^{m+1} = \lambda[\alpha v_{ik} + \beta^0(x_{kPb} - x_{ik}) + \beta^1(x_{iGb} - x_{ik}) + \beta^2(x_{Gb} - x_{ik})] \tag{10}$$

Adjustment factor formula is as follows:

$$\lambda = \frac{2}{2 - \phi - \sqrt{\phi^2 - 4\phi}} \qquad \phi = \beta^0 + \beta^1 + \beta^2 \tag{11}$$

In the formula, m is the number of iterations, i is the number of particle, k is the number of subgroups, x_{kPb} is the optimal position of the k-th particle, x_{iPb} is the optimal position of the k-th subgroup after the m-th iteration, x_{Gb} is the optimal position of the whole Particle Swarm after the m-th iteration

In this way, under the action of the adjustment factor, the range of particle trajectories in the algorithm subgroup increases, the particle follows closely the flight of the optimal particle by adjusting, the overall convergence of the algorithm will be greatly improved.

4. Resource Scheduling Strategy in Cloud Environment

4.1. Coding Strategy

Based on the characteristics of cloud environment resource scheduling, the particle coding method is proposed using decimal coding, the number of resource R is m, the number of task is n, the number of particle in Particle Swarm is k, among them, $x_i = \{x_{i1}, x_{i2}, \cdots x_{in}\}$, $i \in [1,k]$, in this way, express task scheduler to run on resource x_{ij}, the initial position of particle is obtained randomly in $[1,m]$, the encoding of the particle is: （3,2,5,4）， this means that the 4 tasks correspond to the number of the used resources, that is, $t_1 \to x_i[3]$, $t_2 \to x_i[2]$, $t_3 \to x_i[5]$, $t_4 \to x_i[4]$.

4.2. Fitness Function

In the iteration of the algorithm, the position of the next generation particle is determined by the value of the fitness function. Cloud computing scheduling strategies need to consider two factors: time and cost. The focus of this paper is investigated the completion time of the task, that is, the higher the fitness value, the better the properties of particle, the greater the possibility of the optimal solution is obtained. Based on this model, the fitness function of Particle Swarm is defined as follows:

$$f(i) = \frac{1}{\max_{j=1}^{m} \sum_{i=1}^{k} S(t_i)} \tag{12}$$

4.3. Algorithm Step Analysis

Step 1: Initialize the population maximum iterations and adjust population size and other parameters;

Step 2: Divided Particle Swarm, subgroup divided into uniform in size;

Step 3: According to the fitness function to calculate the fitness value of particle, determine the global optimal, sub optimal and individual optimal;

Step 4: Determine the state of the particle in the subgroup, determine whether the implementation of the Cauchy distribution;

Step 5: Iteration number plus 1, update the velocity and position of the particle;

Step 6: Judge whether the algorithm meets the termination conditions. If satisfied, then the end and the output of the optimal solution, otherwise return to step 4.

5. Experimental Analysis

The experiment analyze the performance of the improved algorithm by using CloudSim platform, which include two aspects: on the one hand is the platform test, including the SLA violation rate, resource consumption and different task time ratio, On the other hand, the performance of Particle Swarm Optimization (PSO) itself is analyzed, the convergence of the algorithm is mainly analyzed by benchmark function.

Three Particle Swarm Optimization algorithms are analyzed in the experiment, standard Particle Swarm Optimization algorithm, PSO; the improved Quantum-behaved Particle Swarm Optimization algorithm, QPSO; this paper proposed the multi-group parallel Particle Swarm Optimization algorithm, IMPSO. The size of the whole population is 40, the subgroup size is 10, the maximum number of iterations is 400. First of all, to analyze the platform SLA violation rate and platform consumption, the following list shows the results:

Table 1. SLA Violation Rate

days	PSO	QPSO	IMPSO
1	35.25	38.44	40.98
2	34.18	36.35	37.83
3	33.36	34.96	35.14
4	32.72	33.13	33. 05
5	32.21	32.18	31.52
6	31.78	31.34	30.16
7	31.42	30.65	29.05
8	31.16	30.13	28.07

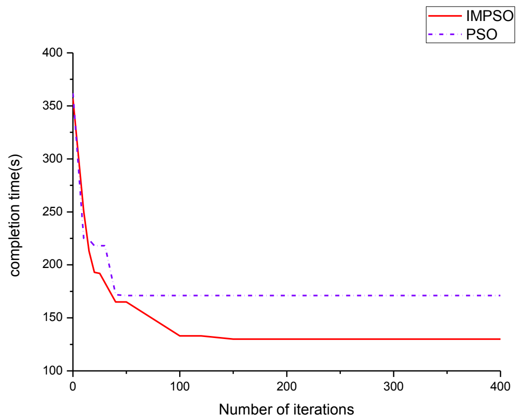
Table 2. Platform Consumption

days	PSO	QPSO	IMPSO
1	1213.56	1334.83	1388.46
2	1832.76	1915.65	1955.26
3	2457.17	2574.98	2512.93
4	2969.71	2981.62	2903.58
5	3479.08	3454.19	3374.59
6	4090.24	4011.48	3920.38
7	4713.54	4679.12	4544.71
8	5319.89	5202.44	5057.85

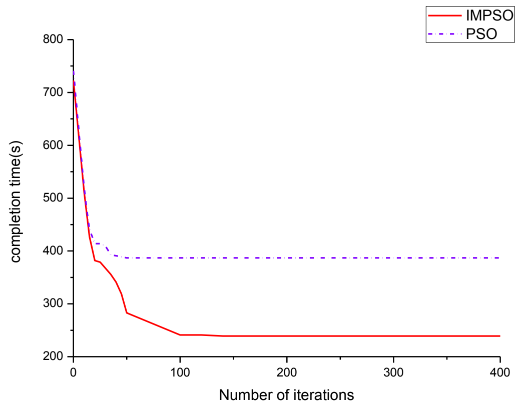
From the data analysis in Table 1 is known that the best performance in the early test is not the IMPSO algorithm, but the PSO, which is because of the coordination problem of the different subgroups at the early stage of algorithm, with the deepening of the algorithm, the optimal solution of the subgroup is generated, and it can be seen that the violation rate of IMPSO decreases steadily, while the other two algorithms have no obvious.

By comparing the data in Table 2 found that the initial value of the three algorithms has little difference. With the extension of time, consumption increased simultaneously, and the other two algorithms increased significantly, especially the Unimproved PSO algorithm. In the later period, the performance of IMPSO algorithm is reflected, its consumption rises slowly and smoothly.

In order to investigate the task scheduling efficiency of optimization algorithm, the experiment is compared through the consumption of the total time, respectively test the consumption time of the two groups data, the number of task respectively are 200 and 400, the following is the comparison chart:



(a) Comparative Analysis of Fixed Number of Task is 200

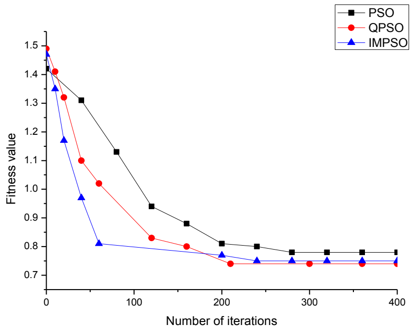


(b) Comparative Analysis of Fixed Number of Task is 400

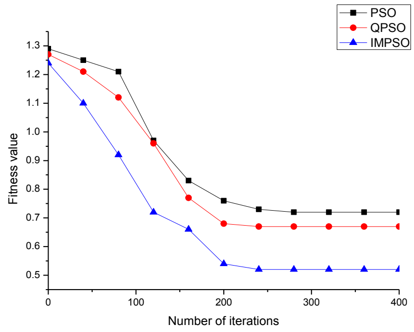
Figure 1. Time Consumption Comparison Chart

From the data analysis in Figure1 shows that the improved IMPSO algorithm has better convergence, and its time consumption is the least. The convergence of the PSO algorithm in the graph is the fastest, when the number of iterations reaches 40, the convergence is complete. In contrast, IMPSO complete the convergence in the 120 time, but the difference in the time spent between the two is nearly 50 seconds. In Figure2, the number of tasks is 400, compared to the three, the number of iterations is more close, the convergence of IMPSO is also good, however, compared with (a), it can be found that with the increase in the number of tasks, the time-consuming difference between the three algorithms is widen. Compared with the standard PSO algorithm, IMPSO saves about 150 seconds, which shows that with the increase of the number of tasks, the improved algorithm can effectively avoid the local constraint, more conducive to obtain the global optimum, the platform efficiency has been significantly improved.

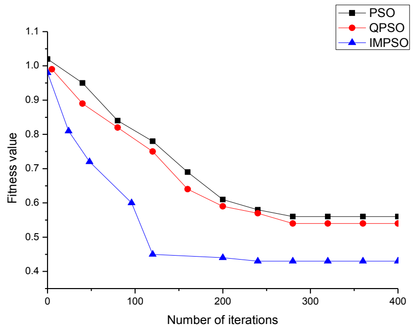
For an investigation on convergence of the algorithm, the paper intends to use three benchmark functions to implement the comparative analysis, the corresponding comparison chart is as follows:



(a) Convergence Comparison



(b) *Rosenbrock* Convergence Comparison



(c) *Griewank* Functional Convergence Results

Figure 2. Comparative Analysis of Benchmark Functions

From the data analysis in Figure2 shows that the comparison of the benchmark functions corresponding to the three algorithms, the convergence of IMPSO is slightly lacking in Figure(a), the other two groups showed better performance, especially in the last group, the convergence rate and function values are the best. It can be seen from the figure that with the increase of the number of iterations, the performance of the improved algorithm becomes more and more obvious, which is mainly due to the improvement of the local optimal constraint in the algorithm, the performance of the particle is judged in each iteration, the particle with poor performance are improved, and the subgroup strategy also accelerates the convergence of the algorithm, this can also effectively improve the accuracy of the algorithm.

6. Conclusion

Resource scheduling in cloud environment is the complex NP problem, using the traditional Particle Swarm algorithm to solve the problem, because the defects of the algorithm itself can't fully reflect the advantages of the platform. Therefore, the multi-group parallel Particle Swarm Optimization algorithm is proposed. By dividing the population into groups of equal size and increasing the diversity of the particle, the particles in the subgroup get their optimal position by using the modified algorithm and implement the fusion. At the same time, to ensure the convergence of the particle with disturbance the poor performance of the particle. In this way, the improved algorithm can greatly improve the efficiency of the cloud computing platform.

References

[1] N. P. Sriram and J. H. Zygmunt, "A practical, secure, and verifiable cloud computing for mobile systems", Procedia Computer Science, vol. 34, **(2014)**, pp. 474-483.
[2] A. N. Gullhav and B. Nygreen, "A branch and price approach for deployment of multi-tier software services in clouds", Computers & Operations Research, vol. 75, **(2016)**, pp. 12-27.
[3] N. Mansouri, "Adaptive data replication strategy in cloud computing for performance improvement", Frontiers of Computer Science, vol. 10, no. 5, **(2016)**, pp. 925-935.
[4] A. Bala and I. Chana, "Prediction-based proactive load balancing approach through VM migration", Engineering with Computers, vol. 32, no. 4, **(2016)**, pp. 581-592.
[5] A. Mohammad, K. Nasrin, M. Sina, M. F. Fatemi and K. Touraj, "A comparative study of time management and energy consumption in mobile cloud computing", 2014 5th IEEE Control and System Graduate Research Colloquium, Shah Alam, Malaysia, **(2016)**, pp. 199-203.
[6] L. Kangchan, P. Chulwoo and Y. H. Dong, "Development of service verification methodology based on cloud computing interoperability standard", International Journal of Smart Home, vol. 7, no. 5, **(2013)**, pp. 57-66.
[7] M. A. Kaleem and P. M. Khan, "Commonly used simulation tools for cloud computing research", International Journal of Smart Home, 2015 International Conference on Computing for Sustainable Global Development, New Delhi, India, **(2015)**, pp. 1104-1111.
[8] M. Haydee and W. Junzo, "Gaussian-PSO with fuzzy reasoning based on structural learning for training a Neural Network", Neurocomputing ,vol. 172, **(2015)**, pp. 405-412.
[9] C. J. Zhang, Y. J. Yang, Z. W. Du and C. Ma, "Particle swarm optimization algorithm based on ontology model to support cloud computing applications", Journal of Ambient Intelligence and Humanized Computing, vol. 7, no. 5, **(2016)**, pp. 633-63.
[10] T. Kerdphol, Y. Qudaih and Y. Mitani, "Optimum battery energy storage system using PSO considering dynamic demand response for microgrids", International Journal of Electrical Power & Energy Systems, vol. 83, **(2016)**, pp. 58-66.
[11] M. G. Epitropakis, V. P. Plagianakos and M. N. Vrahatis, "Evolving cognitive and social experience in Particle Swarm Optimization through Differential Evolution: A hybrid approach", Information Sciences, vol. 216, no. 24, **(2012)**, pp. 50-92.
[12] R. P. Parouha and K. N. Das, "DPD: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints", Expert Systems with Applications, vol. 63, **(2016)**, pp. 295-309.
[13] M. Helbig and A. P. Engelbrecht, "The Effect of Quantum and Charged Particles on the Performance of the Dynamic Vector-evaluated Particle Swarm Optimization Algorithm", Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, New York, USA, **(2015)**, pp. 25-32.
[14] M. Seyedali and L. Andrew, "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization", Swarm and Evolutionary Computation, vol. 9, **(2013)**, pp. 1-14.
[15] T. Ghazaleh, L. C. Kiong and C. L. Teck, "A novel hybrid SP-QPSO algorithm using CVT for high dimensional problems", Springer Proceedings in Mathematics and Statistics, vol. 95, **(2013)**, pp. 347-354.
[16] S. Mukhopadhyay and S. Banerjee, "Global optimization of an optical chaotic system by Chaotic Multi Swarm Particle Swarm Optimization", Expert Systems with Applications, vol. 39, no. 1, **(2012)**, pp. 917-924.
[17] M. Petrovic, M. Mitic, N. Vukovic and Z. Miljkovic, "Chaotic particle swarm optimization algorithm for flexible process planning", International Journal of Advanced Manufacturing Technology, vol. 85, no. 9-12, **(2016)**, pp. 2535-2555.

Authors

C. N. Zhang, received the Master's degree in computer application technology from University of Science and Technology Liaoning, in 2007. Currently, she is a lecturer at School of Software Engineering at University of Science and Technology Liaoning. Her research interests include Distributed computing and data mining.

Y. R. Li, received the Master's degree in computer application technology from University of Science and Technology Liaoning, in 2008. Currently, he is a lecturer at School of applied technology college at University of Science and Technology Liaoning. His research interests include Distributed computing and data mining.