Effect of Inertia Weight ω on PSO-SA Algorithm

http://dx.doi.org/10.3991/ijoe.v9iS6.2923

Shigang Wang, Fangfang Zhou, Fengjuan Wang
Qiqihar University, Qiqihar, China

***Abstract*—Since particle swarm algorithm was proposed, because of its easy to understand and implement, the algorithm has been rapid development. However, the algorithm is easy to convergence, and the simulated annealing algorithm has strong local search ability, which can make the search process to avoid falling into local optimal solution. Because of the complementary of the advantages and disadvantages of the two algorithms, a new PSO-SA algorithm appears. This paper focuses on researching the effect of inertia weight of PSO-SA algorithm on the performance of the algorithm. PSO-SA algorithm is applied in solving the shortest path on curved surface, through an example to illustrate function of the inertia weight in algorithm.**

***Index Terms*—PSO-SA algorithm, Iterative search, Inertia weight, The shortest path on curved surface.**

I. INTRODUCTION

PSO [1] and SA algorithm [2] can solve many practical problems alone, but they all have their own shortcomings. The PSO algorithm in a short period of time, most particles converge to a local extreme point in the solution space. Therefore, the biggest drawback of the PSO algorithm is easily trapped into local optimal solution. The SA algorithm can accept greater energy values in the iterative process, namely the poor solutions, then the algorithm is a global optimization algorithm and easy to obtain the global optimal solution. The most importance is that the SA algorithm is easy to jump out the local extreme point [3]. Given the complementary characteristics of the two algorithms and the effective improvement of the two algorithms, the new PSO-SA algorithm is obtained and the performance of the algorithm is improved.

The inertia weight has a large effect on the performance of PSO-SA algorithm. The inertia weight with the characteristic of maintaining inertial motion of particles and expanding the of search space, its value can reflect the change of particle velocity. Value size of inertia weight is closely related to searching ability of particle [4]. The larger value of the inertia weight is the larger particle velocity is. Particle would have a larger step size to perform global search. Its global optimization ability is strong and local searching ability is weak. The smaller is the smaller step size of particle is. The particle will tend to fine local search and search in the near, and local search ability is strong. If the inertia weight parameter is too large, the particle swarm may miss the optimal solution, which leads to the algorithm does not converge, or fail to converge to the optimal solution.

In this paper, the value of inertia weight is obtained in PSO-SA algorithm through experiments, which will get the shortest distance between two points in the optimal value of the PSO-SA algorithm applied to the shortest path problem in the curve surface. The example proves that the value of inertia weight is optimal.

II. PSO-SA ALGORITHM

A. Basic Particle Swarm Algorithm

PSO Algorithm is initialized to be a group of random particles (random solutions), then get the optimal solution by iteration. In each iteration, particles updated themselves by tracking two extreme values to find the optimal solution which is called the individual extreme value *pbest*. Another extreme value is the current optimal solution of the entire population which is called global extreme value *gbest*. It also uses the whole population but only one portion as the neighbors of the particles. Then all neighbors of extreme value were the local extremum. When finding the optimal value, the velocity and position of each particle was updated according to the following formula

$$V_{K+1} = W \cdot V_K + C_1 \cdot \text{rand}(pbest_K - X_K) + C_2 \cdot \text{rand}(pg\_plus_K - X_K) \quad (1)$$

$$X_{K+1} = X_K + V_{K+1} \quad (2)$$

Where: V_K is the particle velocity vector; X_K is the current position of the particle; $pbest_K$ is the position of the optimal solution; C_1, C_2 are the population learning factor; pg_plus_K is the individual which is chosen by roulette rules.

In particle swarm optimization algorithm, particle velocity is limited to a little range [5], but the new location would still become so bad possibly, which will cause slow convergence. So the updated position needed to be limited.

B. Improved Particle Swarm Optimization

The point of departure for the simulated annealing algorithm applied to the optimization problem is based on the similarity of the physical annealing process of solid material and general optimization problem. The basic idea of the algorithm is that starting from a given solution, randomly generating another solution from the neighborhood, accepting a new solution by a certain probability, accepting the objective function deteriorating in a limited range allowed by criteria [6]. Specific contents are modified as follows.

Accept the objective function deteriorating in a limited range allowed by criteria, but do not make a choice according to probability, directly according to $\Delta E < e$, which is the deteriorating range allowed by objective function. The specific algorithm is shown as follows.

Step 1: Initialize each particle. Set the number of particles is *N*, randomly generate initial solutions and *N* initial velocities;

Step 2: Generate the new location of particles according to the current position and velocity;

Step 3: Calculate the fitness value of the new location of each particle;

Step 4: If the particle's fitness value is better than the original individual extreme, set the current value as *pbest*;

Step 5: Find the global extremum according to each particles individual extreme;

Step 6: Update their own velocity by formula (1);

Step 7: Update their own location by formula (2);

Step 8: Calculate the length of the path generated by the new location, if the new length is less than the former length, change the new path to the old path.

Step 9: Return Step 2, calculate the next population until the repeat number is greater than the alternation number *N*.

III. VALUE OF INERTIA WEIGHT ω

The inertia weight ω is one of the important parameters of PSO-SA algorithm, whose choice is related to the balance between local and global search ability, affecting the convergence performance of algorithm [7], an appropriate value ω use the least number of iterations to find the optimal solution. Inertia weight value ω is determined by the formula

$$\omega=\omega_{max}-\frac{\omega_{max}-\omega_{min}}{iter_{max}}\cdot iter \tag{3}$$

Among them, ω_{max} and ω_{min} stand for the starting and ending values; in the particle swarm algorithm, $iter_{max}$ and $iter$ represent the maximum number of iterations and the number of iterations. In the formula (3), the inertia weight value ω and the maximum number of iterations $iter_{max}$ is inversely proportional. With the increasing of $iter_{max}$, value of inertia weight decreases. So in the early operation, the algorithm's global search ability is strong, and in the operation period, local search is more and more refined.

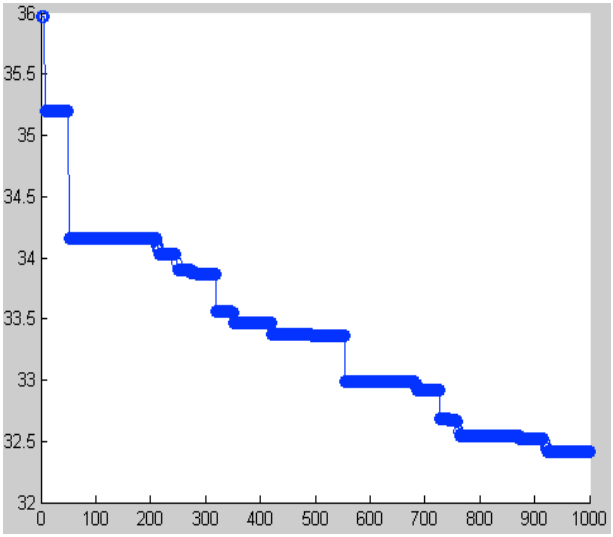
In order to determine the effect of inertia coefficient on the performance of the algorithm, Shi and Eberhart designed the following experiment [8].

To compare the effects of ω and V_{max} two parameters on algorithm performance and analyze different status of both the algorithm, Shi et al. Detailed observe the operation results of different values ω and V_{max}, and add up the number of iterations required in the course of 30 run, some experimental data are shown in table I.

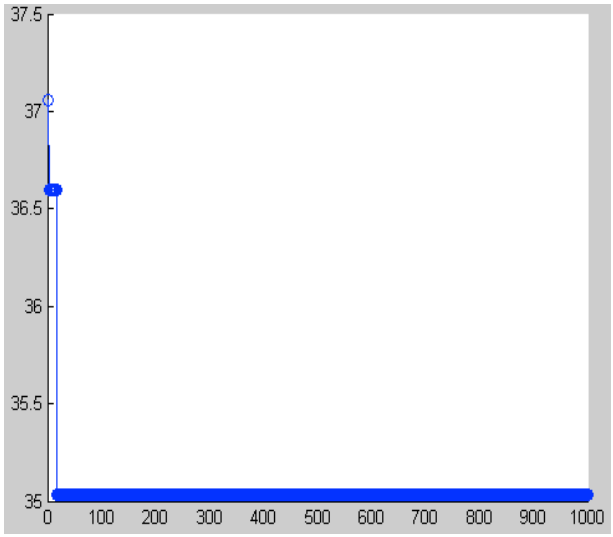
Through the analysis of the experimental data, we conclude: when $V_{max}\geq 3$, and $\omega\in(0.9,1.2)$, it can reach ideal results in general. A large number of experiments have found value ω in [0.9, in the range of 1.2] algorithm has better performance [9]; Fig. 1 shows the performance of algorithm in different values of convergence. However, for different solving problem, the inertia weight ω is not the same [10]. The inertia weight ω in this research is mainly concentrated in the range of 0.9~1.2 near, to run algorithm with MATLAB, when the algorithm of other variables remain unchanged, by changing the value ω to obtain the optimal solution algorithm, and determine the best value ω. In the procedure, when ω values were 0.8, 0.9, 1, 1.1, 1.2, 1.4. Convergence curve is obtained in Fig.1 shows.

TABLE I.
THE AVERAGE ALGEBRAIC ALGORITHM IN DIFFERENT ω AND V_{max} CONVERGENCE

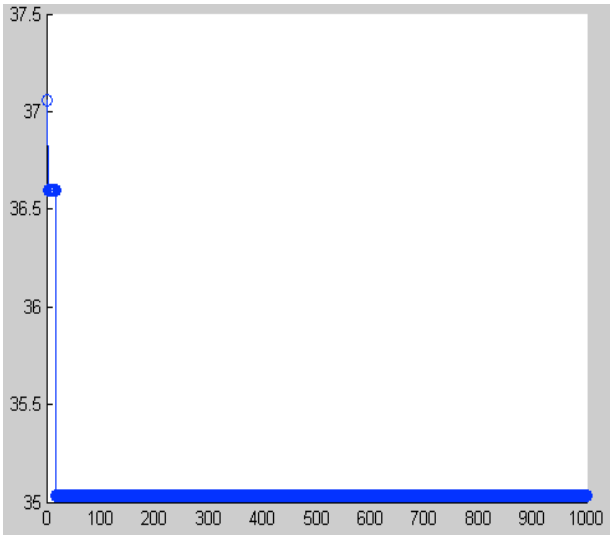
	ω								
V_{max}	1.4	1.2	1.1	1	0.9	0.8	0.7	0.6	0
3	2387	1804	1758	1653	738	438	402	356	663
4	—	2215	2128	1967	946	439	471	421	652
5	—	—	—	2456	1090	366	503	535	1133
10	—	—	—	658	2027	460	789	490	895
X_{max}	—	—	—	—	—	974	879	927	933



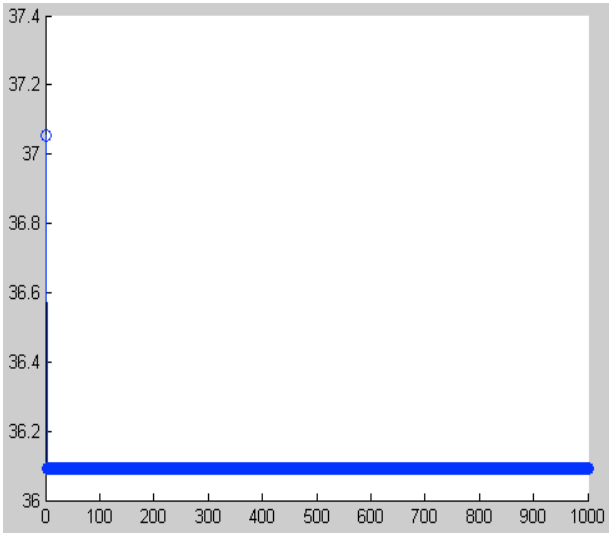
(a) ω=0.8



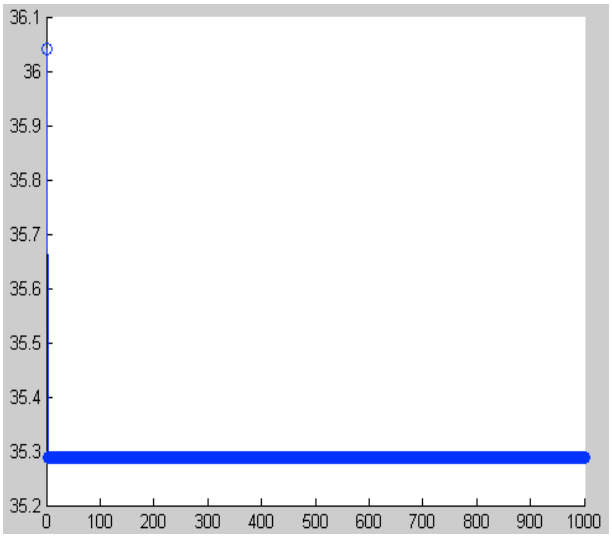
(b) ω=0.9



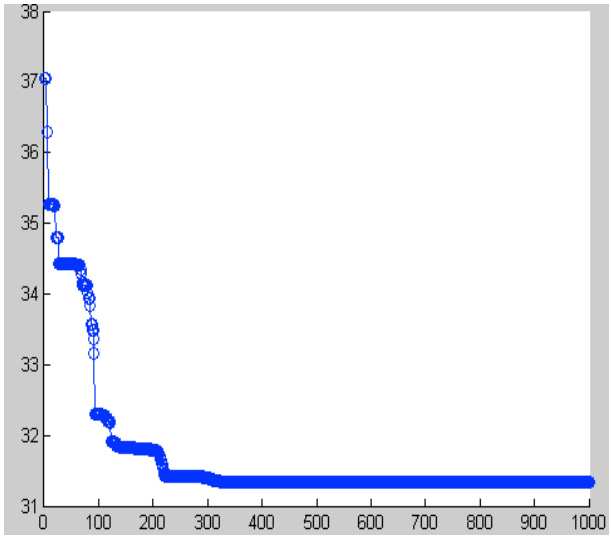
(c) ω=1.0



(d) ω=1.1



(e) ω=1.2



(f) ω=1.4

Figure 1. Convergence curve

From the above convergence map can be seen, when ω=0.8, the convergence performance is not stable. When ω=0.9, the convergence of the algorithm is much better. With increasing of the inertia weight ω gradually, the convergence of the algorithm gradually stabilized. When ω=1.05, the convergence of the algorithm is the best and convergence curve is a smooth straight line. When ω=1.05, with increasing of ω, convergence curve becomes unstable. When ω=1.4, the convergence of the algorithm also becomes very poor. This in turn has also proved that, when the range of inertia weight ω between 0.9 and 1.2, the performance of the algorithm is good and optimum solutions are obtained.

IV. ALGORITHM TESTING

Set the special surface

$$z=20(\sin(x^2+y^2))/(x^2+y^2)+10(\sin((x-3)^2+(y-5)^2))/((x-3)^2+(y-5)^2)+$$
$$15(\sin((x-1)^2+(y-3)^2))/((x-1)^2+(y-3)^2))+$$
$$18(\sin((x+1)^2+(y+3)^2))/((x+1)^2+(y+3)^2)$$

as an example.

Calculate the shortest path between two fixed points A(-7.0, -5.0) and B(7.0, 5.0) in the process of circle arc approach, give an auxiliary parameter, coordinates of basic points are set according to the auxiliary elliptic equation and species individual count. All the parameters in this simulation are set as follows: N=40, ω=1.05, C_1=C_2=2, number of node in each path is 8 (including the starting point and end point), iteration number M=1000. In the end, the shortest length of optimized path is 37.0553.

Convergence curves in this simulation are shown in Fig. 2.

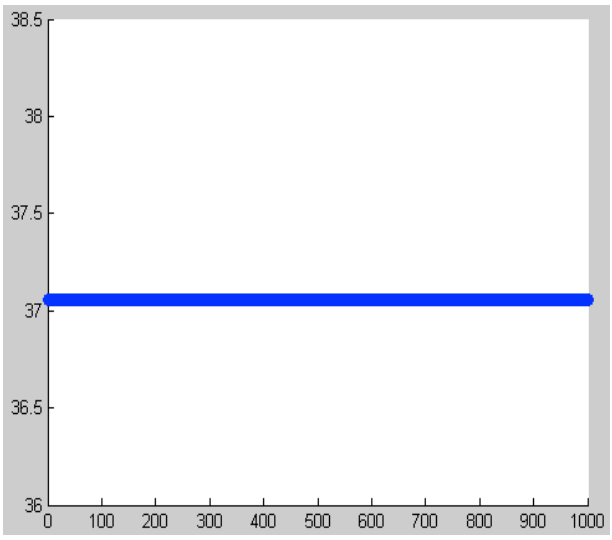


Figure 2. Convergence points

Fig. 3&4 showed the initial path set. Fig. 5&6 showed the optimal path with circle arc approach method.

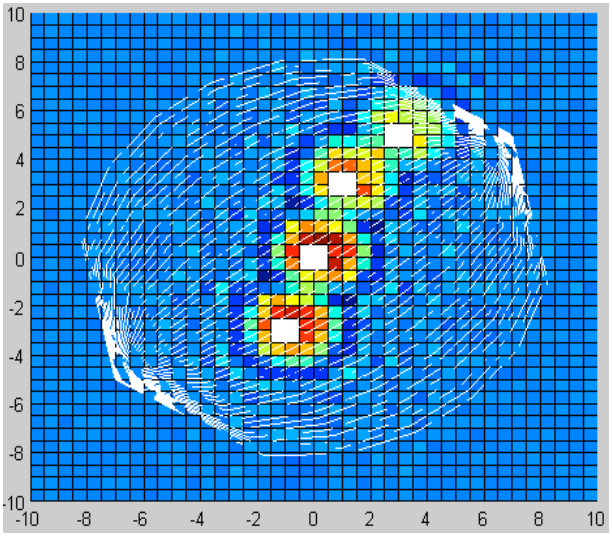


Figure 3. The two-dimension plan of initial path set

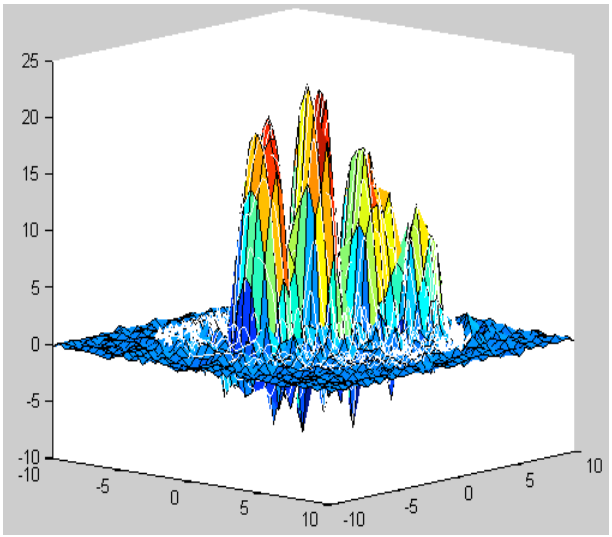


Figure 4. The three-dimension plan of initial path set

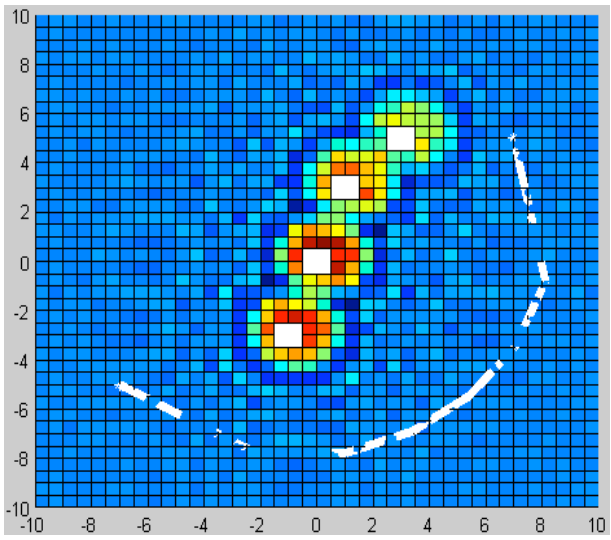


Figure 5. The two-dimension plan view of the optimal path

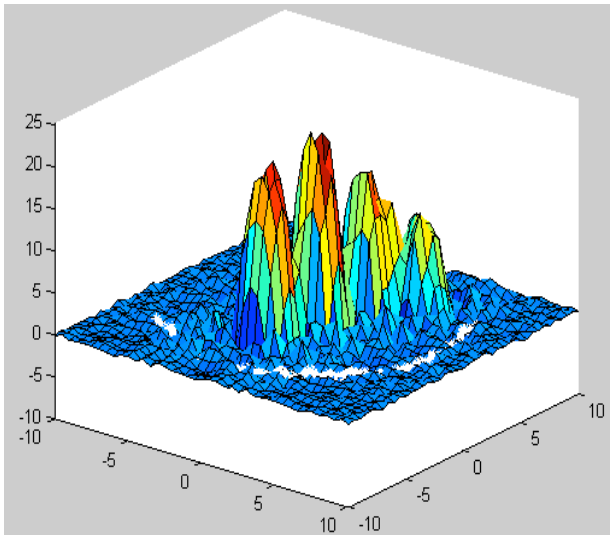


Figure 6. The three-dimension plan view of the optimal path

The node sequence of optimal path after calculating is shown in table II.

TABLE II.
NODES SEQUENCE

X	-7.000	-1.8786	1.0297	3.4144	5.4143	7.0297	8.1214	7.000
Y	-5.000	-7.9414	-7.7845	-6.8944	-5.4658	-3.4987	-0.7986	5.000
Z	-1.682	1.4012	1.1773	-1.2861	0.3774	0.0720	-1.0577	-2.2755

V. CONCLUSION

The PSO-SA algorithm needs fewer parameters adjustment, easier programming, which can effectively solve the problem of different surface optimization. This paper proves that the inertia weight ω has great effect on the algorithm. Selecting the appropriate value of ω is very necessary. The appropriate value of ω can make the algorithm have a good convergence. The examples of numerical simulation in this paper shows the PSO-SA algorithm has better convergence performance when

ω=1.05. It can obtain the ideal solution when the algorithm is used in the free surface to solve the shortest path between two points.

REFERENCES

[1] Kennedy, Eberhart, "Particle swarm optimization", *Proceedings of IEEE International Conference on Neural Networks Perth*, (1995), pp. 1942-1948. http://dx.doi.org/10.1109/ICNN.1995.488968

[2] Kirkpatrick S, Gelatt C., VecchiM., "Optimization by Simulated Annealing", *Science*, No. 220 (1983), pp. 671-680. http://dx.doi.org/10.1126/science.220.4598.671

[3] Shi Liping, "The simulated annealing algorithm and improved study", *Information Technology* , No. 2 (2013), pp. 176-178.

[4] Dong Pingping, Gao Donghui, Tian Yubo, "An improved adaptive inertia weight particle swarm optimizational gorithm", *Computer Simulation*. Vol. 29, No. 12 (2012), pp. 283-286.

[5] Tsai S J, Sun T Y., "An improved multi-objective particle swarm optimizer for multi-Objective Problems", *Expert Systems with Applications*, Vol. 37, No. 8 (2010), pp. 5872-5886. http://dx.doi.org/10.1016/j.eswa.2010.02.018

[6] Zhao Jinghe, Xie Ling, "Simulation TSP simulated annealing algorithm based on LabVIEW", *Electronic Design Engineering*, Vol. 19, No. 17 (2011), pp. 31-34.

[7] Wang Dengke, Li Zhong, "The cloud calculation task scheduling algorithm based on particle swarm optimization and ant colony optimization", *Computer Software and Application*, Vol. 30, No. 1 (2013), pp. 290-293.

[8] Shi Y., Eberhart R. C., "A Modified Particle Swarms Optimizer", *Proceedings of the IEEE International Conference on Evolutionary Computation*. Piscataway, (1998), pp. 69-73.

[9] Sun, Ying, Xiong, "Job-shop Scheduling Problem Based on Particle Swarm Optimization Algorithm", *Sensors&Transducers*, No. 16 (2012), pp. 116-127.

[10] Costello Kerry E, Matrangola Sara L, Madigan Michaei L, "Independent effects of adding weight and inertia on balance during quiet standing", *Biomed Eng Online*, Vol. 11, No. 1 (2012), pp. 20-22. http://dx.doi.org/10.1186/1475-925X-11-20

AUTHORS

Shigang Wang is with the School of Mechatronics Engineering, Qiqihar University, Qiqihar 161006, China (e-mail: hljwangsg@ 163.com).

Fangfang Zhou is with the School of Mechatronics Engineering, Qiqihar University, Qiqihar 161006, China (e-mail: 275982345@qq.com).

Fengjuan Wang is with the School of Mechatronics Engineering, Qiqihar University, Qiqihar 161006, China (e-mail: wfj320110@163.com).

This work was supported by the Natural Science Foundation of Heilongjiang Province, China under Grant E201106 and the Project of Department of Education of Heilongjiang Province, China under Grant 12521597. It is an extended and modified version of a paper presented at the 2012 International Conference on Artificial Intelligence and Its Application in Industry Production (AIAIP 2012), held in Wuhan, China in December 2012.Manuscript received 13 May 2013. Published as resubmitted by the authors 26 June 2013.