

Research Article

An Adaptive Particle Swarm Optimization Algorithm Based on Directed Weighted Complex Network

Ming Li, Wenqiang Du, and Fuzhong Nian

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Correspondence should be addressed to Wenqiang Du; 877621050@qq.com

Received 12 November 2013; Revised 22 January 2014; Accepted 21 February 2014; Published 2 April 2014

Academic Editor: Ge Guo

Copyright © 2014 Ming Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The disadvantages of particle swarm optimization (PSO) algorithm are that it is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process. To deal with these problems, an adaptive particle swarm optimization algorithm based on directed weighted complex network (DWCNPSO) is proposed. Particles can be scattered uniformly over the search space by using the topology of small-world network to initialize the particles position. At the same time, an evolutionary mechanism of the directed dynamic network is employed to make the particles evolve into the scale-free network when the in-degree obeys power-law distribution. In the proposed method, not only the diversity of the algorithm was improved, but also particles' falling into local optimum was avoided. The simulation results indicate that the proposed algorithm can effectively avoid the premature convergence problem. Compared with other algorithms, the convergence rate is faster.

1. Introduction

The particle swarm optimization algorithm is a population-based stochastic optimization algorithm proposed by Kennedy and Eberhart in 1995 [1]. To get the optimal solution, through simulating birds foraging behavior, mutual collaboration between individuals and sharing the internal information in particles are investigated. Due to its fast computing speed and the parallel processing, particle swarm optimization algorithm has been successfully applied in many areas such as neural network training, function extremum estimation, and face detection and recognition [2–4].

The disadvantages of particle swarm optimization (PSO) algorithm are that it is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process. A great number of investigations have been done to improve PSO algorithm in the last decades. In [5], the particle optimization based on the different topology structure was proposed. Although it can avoid algorithm falling into local optimum to some extent, however, different problems need to be based on the different topology structure. And it is difficult to predict the best topology in

advance. So the adaptability of the algorithm is not ideal. In [6], the particle swarm algorithm was optimized according to the features of scale-free network. The optimization capability of this algorithm was improved in the high-dimensional space. However, scale-free networks are not necessary for optimal topology. In [7], the multidirectional learning adaptive particle swarm algorithm was proposed. The optimal solution of the entire group to complete the speed is updated by following their own optimal solution and other particles' optimal solution with the same dimension. This leads to an improvement in PSO algorithm which was further improved in solving high-dimensional optimization problem. However, the particle network topology optimization is not considered in the algorithm. The neighborhood network topology of PSO has a great influence on searching the final optimal solution.

Traditional neighborhood structure of PSO is a global coupled network. In the structure, there is a connection between any two pairs of nodes [8]. In an iterative process, each particle will be compared with all other particles (positions) and moves closer to the optimal particle. In all network topology with the same number of nodes, the global coupling network has the smallest average path length ($L = 1$)

and the largest cluster coefficient ($C = 1$) [9]. However, the traditional PSO is easy to fall into local optimum in the high-dimensional space. The ability of optimization is not enough and the diversity of learning between the particles is inflexible in the global coupling network.

In this paper, each particle is only compared with neighbors (position) by using small-world network model to initialize the neighborhood structure of PSO. Therefore, the particle's convergence speed is fast and all particles are able to the most optimal value, and all particles are able to keep looking for optimal performance (i.e., the diversity of particle solution). The neighborhood structure of small-world network model within the acceptable error limits should have a better optimization effect on deep optimal point in functions. When PSO algorithm eventually is convergence, the node's degrees of particles obey power-law distribution [10]. So we can select a kind of scale-free networks as a final neighborhood structure of particles. Based on the above content, an evolutionary mechanism of the directed dynamic network is proposed.

By introducing the characteristics of scale-free network model and small-world network model, based on directed weighted complex network, an adaptive particle swarm optimization algorithm (DWCNPSO) is proposed. The simulation results show that the proposed algorithm, especially in high-dimensional space, has a good performance. The premature convergence problem can be avoided, and the convergence rate in the late iterative process is faster than other algorithms.

2. The Original Particle Swarm Optimization

In the original PSO, the system is initialized as a set of random solutions; each particle is considered to be a potential solution (fitness) which moves in the search space following the optimal particle [11]. Each particle is made up of two n -dimensional vectors, where n is the dimensionality of the search space. These are the position vector $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and the velocity vector $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$, $i = 1, 2, \dots, N$. N represents the number of the particles in the search space. We can evaluate the quality of the particles by calculating the value of fitness. The best position encountered by itself is denoted as $p_b = (p_{b1}, p_{b2}, \dots, p_{bn})$ and the best position encountered by the whole particles is denoted as $p_g = (p_{g1}, p_{g2}, \dots, p_{gn})$. The velocity and position of the particles at the next iteration are updated according to the following equations:

$$x_{i+1} = x_i + v_i, \quad (1)$$

$$v_{i+1} = v_i + c_1 r_1 \otimes (p_b - x_i) + c_2 r_2 \otimes (p_g - x_i), \quad (2)$$

where $r_1 \sim U(0, 1)$ and $r_2 \sim U(0, 1)$ are used to improve the random nature of the PSO. And usually the maximum velocity is set to be half of the length of the search space. $c_1 \in [0, 2]$ and $c_2 \in [0, 2]$ are called acceleration coefficients. The default values of $c_1 = c_2 = 2$ are used commonly. They control how far a particle will move in a single iteration. Recent researches have proven that it might be better to choose a larger cognitive parameter, c_1 , but $c_1 + c_2 = 4$.

The original process for implementing PSO is as follows.

- (1) Initialize the particles with random positions and velocities on n -dimensions in the search space.
- (2) For each particle, evaluate the fitness in n variables.
- (3) Compare the fitness of particle with its p_b . If the current value is better than the p_b , then set p_b equal to the current value.
- (4) Compare the fitness of particle with its p_g . If the current value is better than the p_g , then set p_g equal to the current value.
- (5) Update the velocity and position of the particle according to (1) and (2).
- (6) If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), the algorithm is ended or return to step 2.

3. Adaptive Directed Weighted Complex Network Particle Swarm Optimization Algorithm

In this section, we would like to build the complex network model of particle swarm to further optimize and ameliorate learning styles between the particles.

3.1. The Directed Weighted Complex Network Model of Particle Swarm. Network model of particle swarm is described as follows.

Definition 1. A complex network $G = \{V, E, W\}$, V for the set of vertices, E for the set of the edges, and W for the collection of the edge weights. A is the adjacency matrix of complex networks G , $A = (a_{ij})_{n \times n}$, and a_{ij} shows the connection relationship of the node i and the node j : if there are edges between two nodes, the a_{ij} is equal to 1; otherwise, $a_{ij} = 0$. $W = (w_{ij})_{n \times n}$ for the weight matrix of network edges; w_{ij} is the weight on the edge a_{ij} .

Weighted out-degree of node i is as follows:

$$W_1 D_i = \sum_{(v_i, v_j) \in E} w_{ij}. \quad (3)$$

Weighted in-degree of node i is as follows:

$$W_2 D_i = \sum_{(v_j, v_i) \in E} w_{ji}. \quad (4)$$

Adjacency matrix of the A is as follows:

$$A_{ij} = \begin{cases} 1, & D(x_i, x_j) \leq R, V(f(x_i), f(x_j)) > 0 \\ 1, & D(x_i, x_j) > R, V(f(x_i), f(x_j)) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

connect at the probability p , $0 < p < 1$,

where $D(x_i, x_j)$, $i = 1, 2, \dots, N$, $j = 1, 2, \dots, N$, $i \neq j$, represents the Euclidean distance between the two particles,

R is the Euclidean distance threshold between two particles, $f(x_i)$ is the particle's fitness, and $V(f(x_i), f(x_j))$ is the fitness's subtraction of the particle i and the particle j .

Network edge weight matrix W is as follows:

$$W_{ij} = \begin{cases} V(f(x_i), f(x_j)) & a_{ij} \neq 0 \\ 0 & \text{others.} \end{cases} \quad (6)$$

Normalized network edge weights are as follows:

$$w_{ij} = \frac{w_{ij}}{\sum(w_{ij})} \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, N, \quad (7)$$

where $\sum(w_{ij})$ is the sum of the edge weights; after it is normalized, $0 \leq w_{ij} \leq 1$.

3.2. Adaptive Particle Swarm Optimization Algorithm Based on Directed Weighted Complex Network (DWCN PSO). Considering a dynamic evolution process of the network topology of particles in which small-world networks evolve into scale-free networks, a complex network model mentioned above is introduced into the neighborhood structure of the particle swarm optimization. In the process of iteration, the network topology of particles will change under the condition of the in-degrees of nodes obeying power-law distribution. When particles converge to the optimum, the in-degrees of nodes in the particles network obey power-law distribution characteristics.

The learning styles between particles follow three principles.

(1) *With the Method of the Optimal Neighborhood in the Particle Optimization Process.* Each particle is not only to learn from its own optimal ($pbest$) position but also to learn the neighbors' optimal ($lbest$) position. When the distance between particle i and particle j is very close, such as less than a certain threshold R , the two particles are neighbors to each other. Otherwise at a certain probability p ($0 < p \ll 1$), they are neighbors at this time (to be seen as a virtual neighbor).

(2) *Connect Randomly to Other Particles (Virtual Neighbors).* Particles can effectively jump out of local optimal value when trapped in local optimal value.

At this moment particles at a certain probability p ($0 < p \ll 1$) (to be seen as virtual neighbors) are neighbors when the position between particle i and particle j is greater than a certain threshold R . The particle can effectively jump out of local position when not find the optimal neighbor within the distance threshold R , by randomization operations that link the edge, particles still can quickly jump out of local optimal value.

(3) *Dynamic Learning Factor c_2 Is Introduced into the Process of Learning the Optimal Neighbor.* The flying inertia of the particles is heterogeneous on time, and also it is heterogeneous on the space. This can enhance the diversification of learning between particles and avoid particles falling into local optimum in high-dimensional optimization space.

The way of mutual learning between particles is only to learn from the optimal neighbor position including virtual

neighbors. When a particle i has a lot of neighbors, select the optimal neighborhood as a neighbor to edge. The greater subtraction of fitness of two particles is, the more urgent to study the optimal neighbor particle's position is. So a dynamic learning factor c_2 and a changing inertial weight w are introduced into (1), and improved equation (1) is as follows:

$$v_i \leftarrow w \times v_i + c_1 r_1 (p_i - x_i) + c_2 r_2 (p_l - x_i) \quad (8)$$

$$i = 1, 2, \dots, N,$$

where $w = (w_1 - w_2) \times (T - \text{iter})/T + w_2$; iter is the current iteration number; T is the maximum number of iterations; w_1 and w_2 are the initial value of weight and the final value of weight, respectively; $c_2 = (1 + w_{ij})c_1$; w_{ij} is the normalized edge weights; p_l is the optimal location of its neighbors.

The weighted in-degree ($W_2 D_i$) of particles represents the level that the neighbor particles learn from the particle (position), and the weighted out-degree ($W_1 D_i$) of particles represents the level that the particle studies other neighbor particles (positions). The weighted in-degree ($W_2 D_i$) of nodes in the weighted directed weighted complex network equals n ($n \geq 0$); the weighted out-degree ($W_1 D_i$) equals 1 or 0 (to learn the optimal neighbor particle's position or no optimal neighbor particle's position to learn), when the particle swarm algorithm is convergence; the in-degree of the nodes in the network obeys power-law distribution; the maximum weighted in-degree of particles is greater than a certain threshold K as well as the weighted out-degree of the particle which is 0 (community found). At this time, the particle finds the optimal value. In this change of the topology, each particle only learns its optimal neighbor and the in-degree of particles in the network obeys power-law distribution, which has a feature of great heterogeneity (in-degree distribution) in the network. When the optimal position of particles of the next iteration does not exist in the neighbor position of particles, the random p ensures that the particle can adaptively find the optimal value.

The procedure for implementing the DWCNPSO is given by the following steps.

Step 1. Set the parameters: the learning factors $c_1 c_2$, connection probability p of random edge, and population size N , w_1 for the initial value of inertia weight, w_2 for the final value of inertia weight, and T for the number of particle swarm evolutions.

Step 2. Initialize the particle swarm positions and velocities within a certain range of values.

Step 3. Initialize the network neighborhood of the particle swarm; calculate the fitness $f(x_i)$ of each particle and the Euclidean distance $D(x_i, x_j)$ between particles; build the complex network neighborhood structure of particle swarm in accordance with the complex network model; and calculate the adjacency matrix A and weighted matrix W .

Step 4. Compare the fitness of each particle with its optimal value p_i . If the current value is better than p_i , then reset the

TABLE 1: Simulation results obtained from three methods for four functions.

Function	PSO		SFPSO		DWCNPSO	
	Mean	Best	Mean	Best	Mean	Best
f_1	0.000175	$2.36E - 07$	$4.62E - 07$	$1.51E - 25$	0.00011	$1.09E - 29$
f_2	0.687286	0.012409	$4.98E - 05$	$2.84E - 14$	0.178946	$0.00E + 00$
f_3	0.001259	$9.46E - 06$	$1.04E - 07$	$0.00E + 00$	0.000263	$0.00E + 00$
f_4	0.093436	0.017859	0.035261	$6.08E - 07$	0.000167	$4.50E - 09$

p_i to be value equal to the current value. Similarly, compare the fitness of each particle with its neighbor optimal value p_l . The value p_l is equal to the current value if the current value is better than p_l .

Step 5. If the in-degree of nodes in the network neighborhood obeys power-law distribution, all particles carry out optimization in accordance with (8) and (2); when the values of the velocity and the position are beyond the limits, set the limited value as their values; recalculate and update the adjacency matrix A and weight matrix W .

Step 6. If termination conditions are satisfied, the maximum weighted in-degree of particles in complex networks is greater than a certain threshold K and the weighted out-degree of the particles is 0 (community found). At this time, the particle finds the optimal value, and the algorithm is ended; otherwise, go to step 4.

4. Simulation Results and Discussions

4.1. Simulation Results and Experimental Analysis. The simulation experiments are implemented on the computer with MATLAB R2010a, Windows XP, and Intel Core2 CPU clocked at 2.10 GHz, memory of 2 GB.

Four important functions, two of which are unimodal (containing only one optimum) and two of which are multimodal (containing many local optima, one global optimum or many global optima), are considered to test the efficiency of the proposed methods. The four test functions are as follows.

- (1) The Sphere function f_1 : $\min f(x) = \sum_{i=1}^n x_i^2$, $x_i \in [-100, 100]$. It is an asymmetric unimodal function and the global minimum is 0.
- (2) The Rastrigin function f_2 : $\min f(x_i) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$, $x_i \in [-5.12, 5.12]$, $i = 1, 2, \dots, n$. It is highly multimodal. However, the location of the minima is regularly distributed. It is a fairly difficult problem due to the large search space and many local optimal values. It is a multimodal and nonlinear function with global minimum 0.
- (3) The Griewank function f_3 : $\min f(x) = (1/4000) \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$, $x_i \in [-600, 600]$. It is a continuous and multimodal function. The general overview suggests convex function, medium-scale view suggests existence of local extremum, and finally zoom on the details indicates complex structure of numerous local extremums. The global minimum is 0.

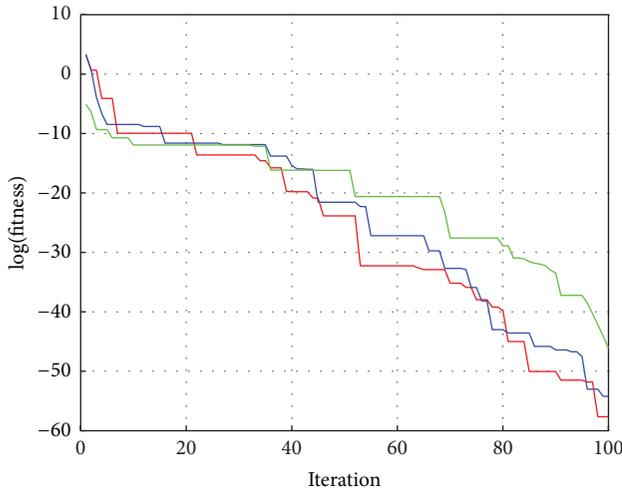
- (4) The Rosenbrock function f_4 : $\min f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$, here, $x_i \in [-30, 30]$, $i = 1, 2, \dots, n$. The global optimum lies inside a long and narrow valley. To find the global optimum is difficult and the function has been used to test the performance of the improved optimization algorithms. The global minimum is 0.

The parameters in the experiments are set as follows: the number of particles is set to 20 and the iteration number is 100. The acceleration factors $w_{\max} = 0.9$ and $w_{\min} = 0.4$, connection probability $p = 0.1$, and learning factors $0 < c_1 < (2w + 2)/(2 + w_{i,j})$ and $c_2 = (1 + w_{i,j})c_1$ are used for the DWCNPSO. And the initial value of x is 0.1 and $y = 0.2$ for the Cat chaotic sequence. The average value and the best value of the function values obtained through 100 simulation runs for four test functions by the proposed PSO, SFPSO, and DWCNPSO algorithms are taken as the measures.

The experimental results are presented in Table 1.

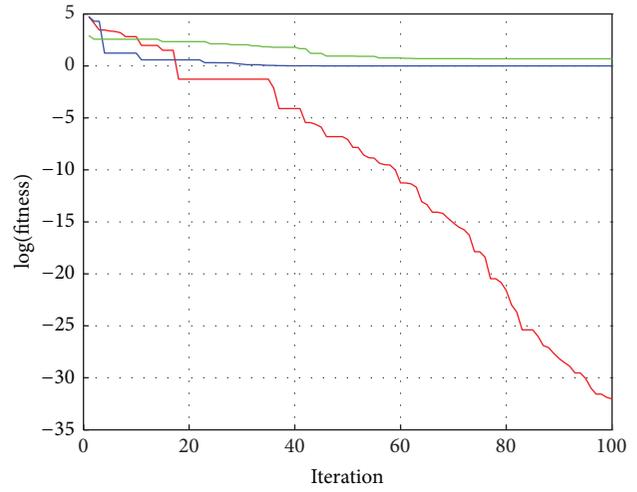
As shown in Table 1, for the Sphere function, it is easy to see that the results obtained by the proposed algorithm are superior to that achieved by the other two algorithms on the average and the best values, respectively. For the Griewank function, Rastrigin function, and Rosenbrock function, the best values obtained by the proposal are superior to the best values of PSO. Although the average value of the proposed algorithm is no better than the average value of SFPSO algorithm, the proposed algorithm is superior to that achieved by SFPSO algorithm on the best values. On the whole, the proposed algorithm is more likely to find the global optimum than other algorithms.

The curves of the evolutionary optimization of three algorithms for four test functions are presented in Figure 1. The best fitness value of test functions is 0. And the accuracy of the fitness value cannot be portrayed accurately when the fitness is close to zero. So taking the e logarithm operation is introduced to resolve this problem. Figure 1 shows that the proposed algorithm is not easy to fall into local optimum and has faster convergence rate in the late iteration. The convergence performance and search accuracy of PSO are the worst in functions f_2 and f_3 . PSO is easy to fall into local optimum and still cannot jump out of the local optimum after much iteration, especially in the performance for Rosenbrock function and Griewank function. Although SFPSO is not easy to fall into local optimum, its search accuracy is not high which is equally efficient with DWCNPSO as well. The convergence performance of the proposed algorithm is equally efficient with other algorithms for the function of Sphere, the Rastrigin function, Griewank function, and



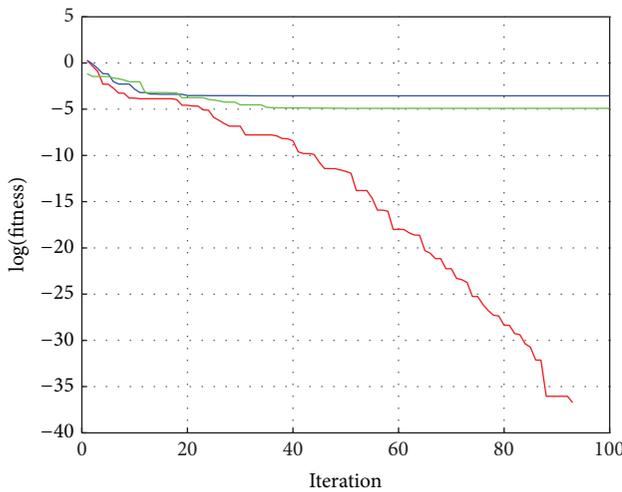
— <DWCNPSO
— SFPSO
— PSO

(a) The evolutionary curve for f_1



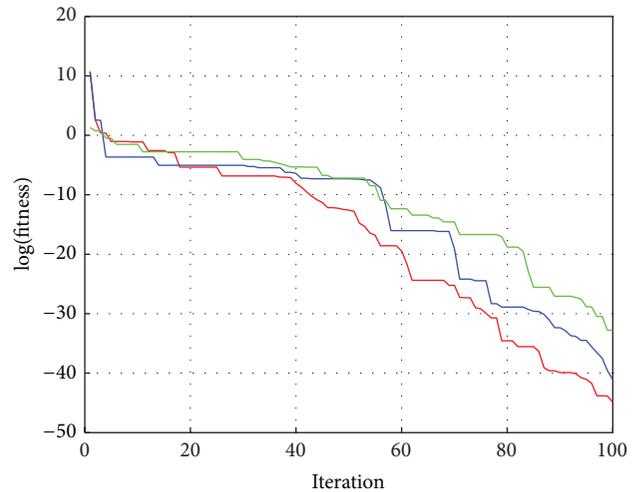
— <DWCNPSO
— SFPSO
— PSO

(b) The evolutionary curve for f_2



— <DWCNPSO
— SFPSO
— PSO

(c) The evolutionary curve for f_3



— <DWCNPSO
— SFPSO
— PSO

(d) The evolutionary curve for f_4

FIGURE 1: The evolutionary curve for three methods for four functions.

Rosenbrock function; the proposed algorithm has the fastest convergence rate, can quickly jump out of local optima after several iterations, and then finds the global optimum.

The Average Degree of Network Impact on DWCNPSO. As mentioned above, the network neighborhood topology of PSO will have an influence on the final optimization. For the same type of network with different properties, a scale-free network, for example, the average degree $\langle K \rangle$ of network, as an indicator will distinguish different topological properties and depict the general structure of the network topology [12].

The following experiment is analyzed based on different average degree of the scale-free network neighborhood topology. Rosenbrock and Rastrigrin functions are tested many times (independent experiment, 500 times), and the test results are shown in Figure 2: when the average degree of $\langle K \rangle$ equals 6.01, the result of optimization is best in Rosenbrock function; when the average degree of $\langle K \rangle$ is 10.23, the particles are easy to fall into the local optimum. For Rosenbrock function, the higher is the average degree of $\langle K \rangle$, the worse is the result of optimization, but, for Rastrigrin function, when the average degree of $\langle K \rangle$ is 10.23, the result of optimization is

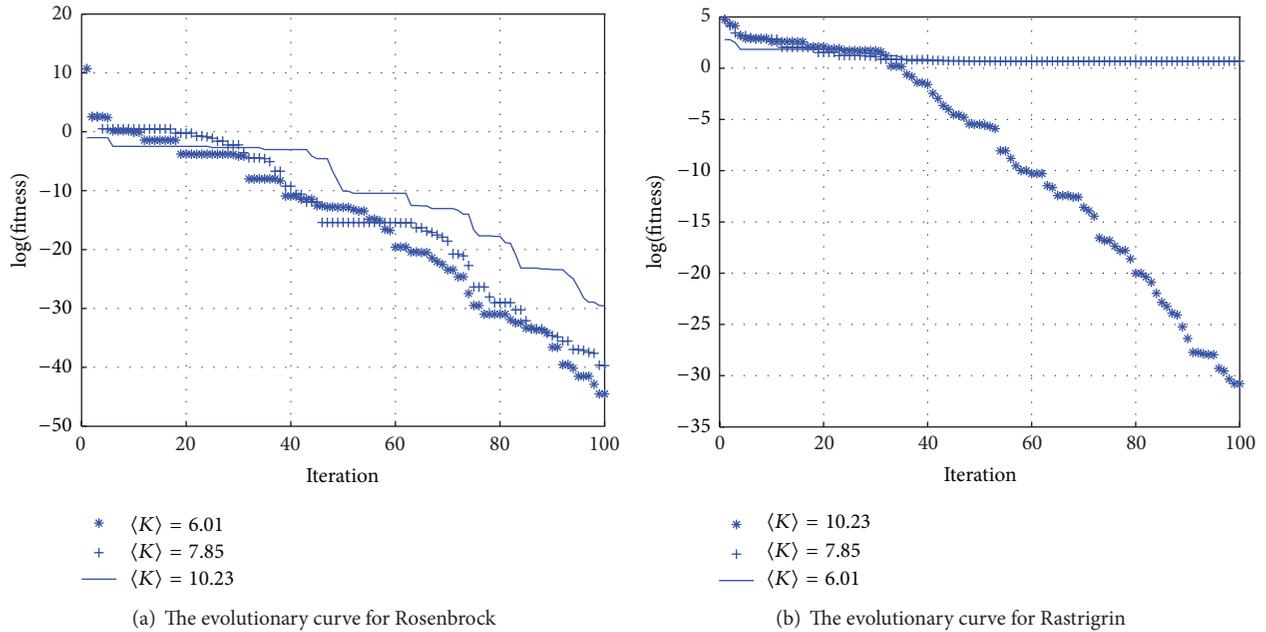


FIGURE 2: The evolutionary curve for three methods for two functions.

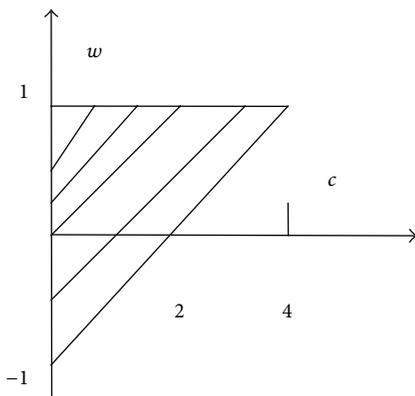


FIGURE 3: The convergence region of parameters of PSO.

best. There will be an optimal range of average degree for two test functions. The optimization effect of the particles will be better within the optimal range of the average degree.

4.2. The Convergence and Computational Complexity Analysis of DWCNPSO

4.2.1. Convergence Analysis of DWCNPSO. The original PSO is convergent, which is proved in literature [13]. As for (1), $w < 1, c > 0, c = c_1 + c_2, 2w - c + 2 > 0$. The convergence region of parameters of PSO is shown in Figure 3. In this paper, the DWCNPSO is proposed by building complex networks and adjusting adaptively learning factor c_2 to improve the original PSO algorithm, which does not change the search mechanism of original PSO. As for (8), $c = c_1 + c_2 = c_1 + (1 + w_{i,j})c_1 = (2 + w_{i,j})c_1, 0 < c < 2w + 2$, when $0 < c_1 < (2w + 2)/(2 +$

$w_{i,j})$, the proposed method is also convergent. This can also be seen from the evolutionary curves for four test functions in Section 4.1.

4.2.2. Computational Complexity Analysis of DWCNPSO.

The maximum iteration is denoted as T . The number of particles is denoted as n . The dimension of the search space is denoted as d . In the original PSO, the computational complexity is $O(2nd + n^2 + 2nd)$. In DWCNPSO, the computational complexity is $O(2nd + n^2 + 2nd + n^2)$. The increased part of computational complexity is caused by building complex networks operation. The disadvantages of particle swarm optimization (PSO) algorithm are that it is easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process. The computational complexity of DWCNPSO is accepted when it is applied to solve the high-dimensional and complex problems.

4.2.3. Performance Analysis of DWCNPSO.

The directed weighted complex network is built on the particle swarm as well as the introduction of dynamic learning factor c_2 and $lbest$ in (1). The flying inertia of the particles is heterogeneous on time, and also it is heterogeneous on the space. The path to study the optimal particle's location is not long and the styles of learning are diverse between particles, when particle swarm is convergence; there are many local extreme points that are not closed in networks and set the best of local extreme point to be value equal to the optimal value of global. If the termination conditions of algorithm are not satisfied and do some disturbances on local extreme points. The effect of disturbance is obvious and the convergence rate is fast.

5. Conclusions

In this paper, an adaptive particle swarm optimization algorithm based on directed weighted complex network (DWC-NPSO) is proposed. By introducing a complex network model and dynamic learning factor c_2 , the flying inertia of the particles is heterogeneous on time and also it is heterogeneous on the space. The diversity of learning between the particles is increased, and the particles can quickly find the optimal solution. When the proposed algorithm falls into local optimum, by randomization operations that link the edge, particles still can quickly jump out of local optimal value. Although some time complexity is increased, however, the number of particles is usually $n < 50$ and the proposed algorithm is advantageous compared with other algorithms. The simulation results indicate that the proposed algorithm can effectively avoid the premature convergence problem and the convergence rate is faster than SFPSO (based on the Cat chaotic mapping) algorithm in the late iterative process.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (no. 61263019), the Fundamental Research Funds for the Gansu Universities (no. 1114ZTC144), the Natural Science Foundation of Gansu Province (no. 1112RJZA029), and the Doctoral Foundation of LUT.

References

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [2] A. W. Mohemmed, M. Zhang, and M. Johnston, "Particle swarm optimization based adaboost for face detection," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 2494–2501, May 2009.
- [3] A. Alfi, "Particle swarm optimization algorithm with dynamic inertia weight for online parameter identification applied to Lorenz chaotic system," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 2, pp. 1191–1203, 2012.
- [4] C. A. Perez, C. M. Aravena, J. I. Vallejos, P. A. Estevez, and C. M. Held, "Face and iris localization using templates designed by particle swarm optimization," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 857–868, 2010.
- [5] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC, USA, 1999.
- [6] C. Yao and J. Yang, "The analysis of the PSO algorithm based on static and dynamic topological neighborhood of SF," *Journal of Chinese Computer Systems*, vol. 33, no. 5, pp. 1113–1116, 2012.
- [7] C. Kan, "Multidirectional learning and adaptive particle swarm optimization algorithm," *Computer Engineering and Applications*, vol. 49, no. 6, pp. 23–28, 2013.
- [8] D. Zhu, X. Zhang, and S. Li, "Discovering fuzzy community structure using local network topology information," *Journal of the University of Electronic Science and Technology of China*, vol. 40, no. 1, pp. 73–79, 2011.
- [9] A. Jiménez, "A complex network model for seismicity based on mutual information," *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 10, pp. 2498–2506, 2013.
- [10] A. Buscarino, M. Frasca, L. Fortuna, and A. S. Fiore, "A new model for growing social networks," *IEEE Systems Journal*, vol. 6, no. 3, pp. 531–538, 2012.
- [11] A. L. Gutiérrez, M. Lanza, I. Barriuso et al., "Comparison of different PSO initialization techniques for high dimensional search space problems: a test with FSS and antenna arrays," in *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP '11)*, pp. 965–969, April 2011.
- [12] C. Tong, J. W. Niu, G. Z. Qu, X. Long, and X. P. Gao, "Complex networks properties analysis for mobile ad hoc networks," *IET Communications*, vol. 6, no. 4, pp. 370–380, 2012.
- [13] S. Kamisetty, J. Garg, J. N. Tripathi, and J. Mukherjee, "Optimization of analog RF circuit parameters using randomness in particle swarm optimization," in *Proceedings of the World Congress on Information and Communication Technologies (WICT '11)*, pp. 274–278, December 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

