International Journal of Control and Automation
Vol.7, No.2 (2014), pp.231-240
http://dx.doi.org/10.14257/ijca.2014.7.2.22

Improved PSO based on Update Strategy of Double Extreme Value

Weidong Ji, Jianhua Wang and Jun Zhang

Department of Computer Science and Information Engineering, Harbin Normal University, Harbin, China

kingjwd@126.com, wjh@vip.163.com, hsd_zj@163.com

Abstract

Particle Swarm Optimization (PSO) algorithm is a new swarmed intelligent optimization technique, which has been widely used to solve various and complex optimization problems, but there are still premature, low precision, slow convergence phenomenon. We proposed an improved PSO based on update strategy of double extreme value by analyzing the updating ways of double extreme. Improved algorithm has good global searching capability through the classical test function, the new algorithm has solutions of high precision, fast convergence, and it is proved that the new algorithm is feasible and effective.

Keywords: *particle swarm optimization, double extreme, precocious, convergence*

1. Introduction

Particle Swarm Optimization (PSO) was proposed by Dr. Eberhart and Dr. Kennedy in 1995 [1, 2], which produced improved and swarmed intelligent algorithm by cooperation and competition between particles. Each individual in PS follow the simple act, follow the example of the experience of the adjacent successful individual, and the cumulative actions unfolded is the best area searching for a high-dimensional space. After the proposing of PSO, it's been many improvements and applications [3, 4]. The simulation results of optimization test was given by Shi when $V_{\max}$ was valued in $[2, X_{\max}]$ separately and inertia factor ω was valued in $[0.1,1.05]$ [5]. Eberhart has suggested that inertia factor ω should be from 0.9 to 0.4 in linear decreasing in searching [6], Shi and Eberhart proposed the way of changing Inertia weight ω by fuzzy technology dynamically and adaptively [7]. Riget and Vesterstroem use the update strategy of ω decreasing with time [8]. Clerc and Kennedy suggested that the value of $(c_1 + c_2)/2$ is 1.494, ω was valued in 0.729 between 0 and 1. They also conducted a parametric stability analysis when PSO algorithm is a linear time-invariant dynamic systems [9]. Robinson and Rahmat-Samii also given another parameter selection, acceleration factor: $c_1 = 2.041$, $c_2 = 0.948$, Inertia factor: $\omega = 0.729$ [10].

Numerous studies indicate that PSO model is easy to premature convergence [11-13], position and velocity of the particle never change, or change very little at these time [14-15]

ISSN: 2005-4297 IJCA

.One reason for rapid convergence is that each particle in the swarm get "Individual cognitive" and "social cognition" capability from the two extremes respectively in the standard PSO iteration, their flight particle velocity and position were changed in this effect. In the complex environments which has a large number of local extrema, g_{best} is easy to fall into local extremum, and PSO itself does not provide any mechanism to make g_{best} get rid of local extrema. So PSO algorithm is easy to premature convergence.

In PSO, the changing of particle position in the searching space is based on successful social psychological tendencies of imitating, the changing of particles in the group subject to its neighbor's experience or knowledge influence. Searching behavior of a particle is influenced by the other particles of the group, particles are close to the optimal solution under the guidance of single extremum and global extremum. In order to improve PSO's global convergence performance, we improve the quality of PSO's quality by the introduction of dual extremum update strategy.

2. PSO with Contraction Coefficient

Clerc introduced the concept of shrinkage factor under PSO, and Speed is a constant contraction, which is called the coefficient of contraction [9].Velocity and position:(1)(2):

$$v_{ij}(t+1)=\chi[v_{ij}(t)+\phi_1 r_1(p_{best(i,j)}(t)-x_{ij}(t))+\phi_2 r_2(g_{best(j)}(t)-x_{ij}(t))] \quad (1)$$

$$x_{i,j}(t+1)=x_{i,j}(t)+v_{i,j}(t+1) \qquad (2)$$

Among them,

$$\chi=\frac{2}{\left|2-\phi-\sqrt{\phi(\phi-4)}\right|} \qquad \phi=\phi_1+\phi_2 \qquad （3）$$

The contraction formula above is got by the formal eigenvalue analysis of Population dynamics model [9].

Contraction coefficient method was developed to a natural dynamic dynamics to promise that the algorithm can converge to a stable point in the case where no speed clamp. When $\phi \geq 4$,Populations can guarantee convergence. Shrinkage factor χ is between 0 and 1,which means speed decreases gradually with time. Literature [9] pointed out, the parameters were taken, $\chi=0.729$,$\phi_1=2.8$, $\phi_2=1.3$.

3. Double Extremum Update Strategy

3.1 individual extreme update strategy

In PSO, each particle updates its velocity and position according to the formula (1) (2),we can see from (1),particle velocity profile is composed of three, namel $\chi v_{ij}(t)$, $\chi\phi_1 r_1(p_{best(i,j)}(t)-x_{ij}(t))$, $\chi\phi_2 r_2(g_{best(j)}(t)-x_{ij}(t))$, Speed profile is shown in Figure 1:

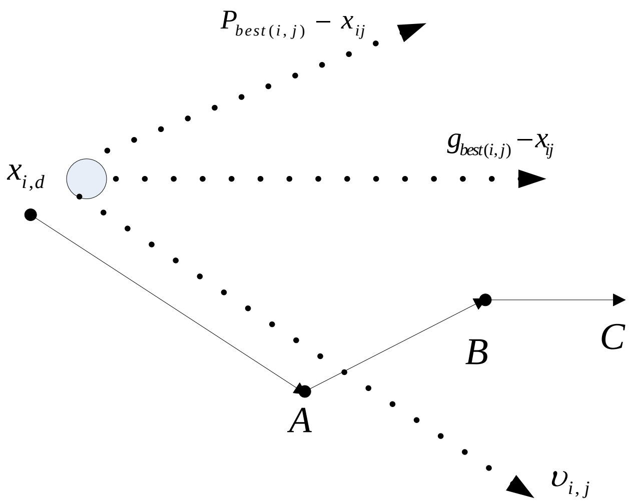


Figure 1. Particle Velocity Updating Diagram

As can be seen from Figure 1, the particles during the update process, firstly, $\chi v_{ij}(t)$ was updated by origin $x_{i,d}(t)$ goes down $v_{ij}(t)$,and reaches point A; $\chi\phi_1 r_1(p_{best(i,j)}(t)-x_{ij}(t))$ was updated by A goes down $p_{best(i,j)}(t)-x_{ij}(t)$,and reaches point B; Finally, $\chi\phi_2 r_2(g_{best(j)}(t)-x_{ij}(t))$ was updated by B goes down $g_{best(j)}(t)-x_{ij}(t)$,and reaches point C. Point C is the new location in conventional PSO's updating. In the updating process, the fitness of A and B are better than C, particles may cross or go beyond the most advantages to the new location. In real foraging process of bird, bird's flight direction is generally zigzag rather than straight, this is also more consistent with birds foraging behavior, who can cover a larger space. After the analysis above we can be found, A, B may be superior to point C in location, but they are missed more, in order to avoid this from happening, we can get the Location and fitness of A and B, and compared with C, we can decide the new location of better, According velocity update formula, speed and position of A,B, and C were already calculated:

$$v_A(t+1)=\chi v_{ij}(t) \qquad x_A(t+1)=x_{i,d}(t)+v_A(t+ \tag{4}$$

$$v_B(t+1)=v_A(t+1)+\chi\phi_1 r_1(p_{best(i,j)}(t)-x_{ij}(t)) \qquad x_B(t+1)=x_A(t+1)+v_B(t+ \tag{5}$$

$$v_C(t+1)=v_B(t+1)+\chi\phi_2 r_2(g_{best(j)}(t)-x_{ij}(t)) \qquad x_C(t+1)=x_B(t+1)+v_C(t+ \tag{6}$$

We choose the better to be the new location, the updating way of $p_{best(i,j)}(t)$:

$$p_{best(i,j)}^{(new)}=\min(f(A),f(B),f(C),f(p_{best(i,j)}^{(old)})) \tag{7}$$

3.2 Cloning strategy of global extreme variability updating

In entire updating iteration of PSO, global extreme plays an important role, which impact on the quality of the optimal solution directly, after the iteration, global extremum often located in or near the area of the most advantages. This design variant clone population updating strategy to change global extreme position vector, to avoid falling into local extremum, improving particle fitness value better, make the global optimum be or near to optimum point as possible. Let mutation probability be p_{pm} ,is generally valued $0.001 \sim 0.1$.Clonal mutation operator [16]:

$$gbest(t) = gbest'(t) + \mu \frac{x_k^{\max} - x_k^{\min}}{2} \tag{8}$$

In this, $\mu = 1 - r^{(1-t/n_t)^5}$, $x_k^{\max}$, $x_k^{\min}$ are searching interval lower limits. r is uniformly distributed between 0 and 1, t is the current iteration, n_t is the maximum number of iterations. Clonal mutation operator added a random value into components of global extreme position vectors, is the copied value of global extreme when iterating. The introduction of μ can promise that mutation size can decrease with the increase of time, which matches with various space in early evolutionary and small space in late variability.

When PSO is being optimization, clone is going in global extreme population according to the mutation probability in iterative process, to be a clone with the same nature as original global extreme $gbest'(t)$,the extreme after cloning were variated, and calculated the fitness, if it is higher than itself, exchange value with each other, or keep them.

By cloning mutation, global population has a self-learning ability extremum, they are able to get rid of the particles, enhance the ability of local extreme, and improve the algorithm efficiency.

4. Performance Test Comparing between Improved Particle Swarm Optimization (IPSO) and GA, PSO.

The four benchmark functions were used to finish Optimization test. Spherical Functions is Unimodal Function; Rosenbrock Function is a morbid function which is difficult to be Minimization; Griewank Function and Rastigrin Function are Multimodal functions with a large number of local optimum points.

$$f_1(x) = \sum_{i=1}^{n-1} x_i^2 \qquad x_i \in [-1\,0\,0\,,\,1](\tag{9}$$

$$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad x_i \in [-100, 100] \tag{10}$$

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1 \quad x_i \in [-600, 600] \tag{11}$$

$$f_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10] \qquad x_i \in [-1\,0\,,\,1] \tag{12}$$

N is the variable dimension, the global minimum of these four test functions are $\min f_i = 0 \ (i = 1, 2, 3, 4)$.

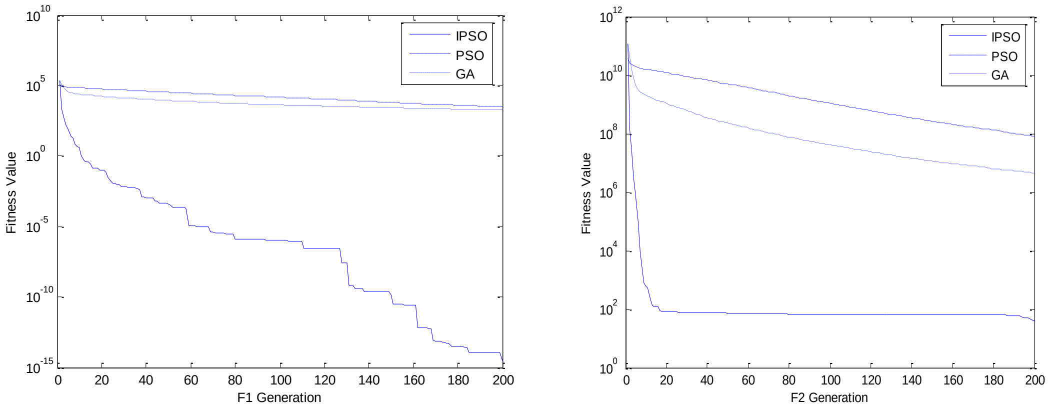
We will have four test by three ways: SGA, PSO, and IPSO, to compare with each others. In this test, parameters: algorithm population size: $P = 30$,SGA has three operators: Crossover probability: $p_c = 0.9$,Mutation probability: $p_m = 0.01$, In PSO, inertia weight factor ω decreases from 0.9 to 0.2 linearly with evolution algebra; Acceleration factor: $c_1 = c_2 = 2$,Maximum speed of particles: $v_d^{\max} = 0.2 x_d^{\max}$,IPSO uses the first way, compression factor: $\chi = 0.729$, $\phi_1 = 2.8$ $\phi_2 = 1.3$,mutation probability: $p_{pm} = 0.01$.Dimension is 80,evolution algebra is 200.After 20 times the three kinds of different algorithms on four benchmark test functions independently, we got standard deviation:

Table 1. Performance Comparison among Three Algorithms

Test Function	Average fitness（Standard deviation）		
	SGA	PSO	IPSO
f_1	1.9133e+003 （498.0693）	3.1336e+003 （2.9493e+003）	2.3276e-015 （9.8526e-015）
f_2	4.4402e+006 （1.2929e+006）	8.1667e+007 （2.7662e+007）	38.0539 （33.8903）
f_3	17.7109 （2.9608）	20.2878 （4.0786）	0 （0）
f_4	611.0205 （42.8763）	624.0736 （92.1739）	5.9508e-015 （1.6647e-014）

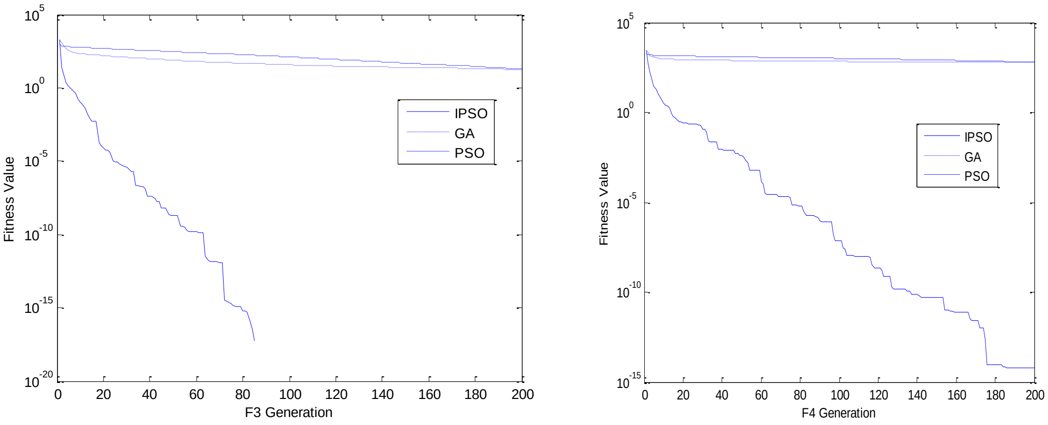
We can see from Table 1, Ipso is better when compared with PSO and SGA, we got the desired global minimum in Griewank Function in IPSO, so IPSO algorithm improves the ability of global searching algorithms significantly. Compared stability of 20 times, the fitness and standard deviation of ISPO are smaller than the others. So IPSO is more stable than the others in the quality of solving.

Figure 2 shows the three algorithms for four benchmark test functions. conditions: Dimension is 40,evolution algebra is 200.Let us run it with 20 times. We can see from Figure 2, IPSO algorithm not only has strong global searching capability, and has fast convergence speed, which can effectively weaken the conventional genetic algorithm and particle swarm optimization algorithm premature convergence, and the phenomenon of easy to falling into local optimal, even for high-dimensional complex functions, the use of this improved PSO algorithm is also achieved better optimization results, which proves this IPSO algorithm.



（a）Benchmark Function f_1

（b）Benchmark Function f_2



（c）Benchmark Function f_3 （d）Benchmark Function f_4

Figure 2. Fitness Value Changing Curve of Three Algorithms on Four Benchmark Functions

5. Static Function Approximation Problem Experimental Study

Adoption feed forward neural network approximation (13) static function:

$$y(x) = \sin(2x)e^{-x} \qquad (13)$$

In the following way to generate sample set: In the interval $(0,\pi]$ we use a pitch of $\Delta = 0.03$ to produce 104 exports to use as the training sample set; In the interval $[0.02,\pi]$ we use a pitch of $\Delta = 0.1$ to produce 32 exports to use as the test sample set.

Assume that the network structure is 1-3-1, Learning Network Adapter function is defined as （14）:

$$E = \frac{\sum_{t=1}^{104}(y_1^d(t) - y_1(t))^2}{104} \qquad (14)$$

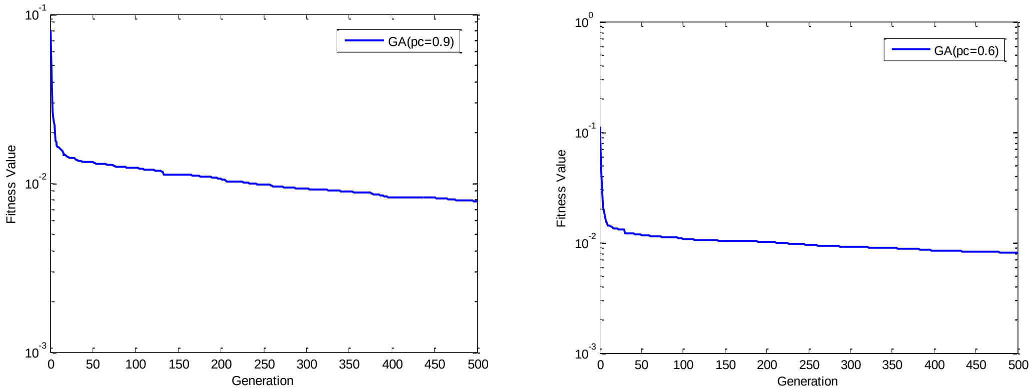
Respectively adoption BP、SGA、PSO and Improved Particle Swarm in this paper to learn Network, Initial weights of the network uniform random generated in the interval $[-3,3]$,the Population size of GA、PSO and Improved Particle Swarm are defined as $p = 30$;In GA algorithm (Nonlinear ranking selection, Arithmetic crossover and Uniform mutation genetic manipulation),the best individual is respectively adopted to conduct experiment in conditions of selection probability is $q = 0.1$, crossover probabilities are $p_c = 0.9$, $p_c = 0.6$. Mutation probability is $p_m = 0.20$ and Evolution algebra is 500.The method of this paper, velocity update adopts the mode of formula(1).Until Compression factor is $\chi = 0.729$,coefficients are $\phi_1 = 2.8$, $\phi_2 = 1.3$. Mutation probability is $p_{pm} = 0.01$, Evolutionary Algorithms $T = 15$, the algorithm terminates. In PSO algorithm, the inertia weight factor ω together with the evolution algebra linear decreasing from 0.9 to 0.4;Acceleration factors are $c_1 = c_2 = 2$;Maximum speed of particles are $v_d^{\max} = 0.2x_d^{\max}$, Termination conditions are the same with GA algorithm. In BP algorithm, learning step is $\eta = 0.50$, friction coefficient is $\alpha = 0.25$,the algorithm is stopped after training 3000 times. Four methods all repeat the test for 10 times, and then to compare the Training set error, Test set error, Time cost and other index of the four algorithms, Test results are shown in Table 2.

As can be seen from Table 2. Although in the 10 experiments, the Improved Particle Swarm in this paper is inferior to BP learning algorithm in Best training error, but in the Worst training error, the Average training error, Average test error, and the average time and standard deviation of CPU, it is superior to the other three methods. From which we can draw that, Improved PSO spends less time, but get higher accuracy over BP, GA and PSO methods. This fully demonstrates the effectiveness of methods to train the neural network.

Table 2. Comparison of Testing Results for Static Function Approximation

Learning methods	Training error			Average test error	Average time（s）	Standard deviation
	Best training error	Worst training error	Average training error			
BP	2.7803E-6	6.9155E-3	1.1918E-3	2.3576E-4	11.42	2.9604E-4
GA（p_c=0.6）	2.7009E-3	1.1907E-2	8.0124E-3	1.9659E-3	34.6	3.1696E-3
GA（p_c=0.9）	4.6738E-3	1.1360E-2	7.8178E-3	1.4971E-3	34.9	1.4971E-3
PSO	8.0639E-4	1.0131E-2	4.0417E-3	5.7706E-4	37.4	3.2421E-3
IPSO	2.0352E-5	9.7018E-4	1.9784E-4	2.6755E-5	6.4	2.8093E-4

Figure 3 shows that the average fitness value evolution curve comparison chart for 10 runs of GA、PSO and IPSO. The evolution algebra are 500 generations of GA PSO, This method in this paper the evolution algebra is15 generations.



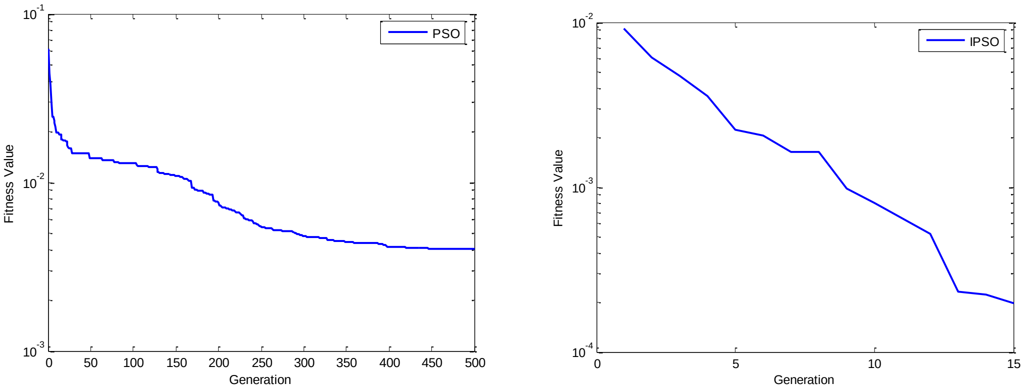


Figure 3. Evolutionary Curve of Average Fitness Value for Static Function Approximation

From Figure 3 we can see this method compared with the other two methods, in the case the Evolution algebra is much smaller than that in GA, PSO algorithm, its adaptation curve declines steeper and even closer to the global optimum compared to GA, PSO algorithm. Explains that the method in this paper to train network convergence speed is much faster and has higher accuracy optimal solution than GA, PSO algorithm.

6. Conclusions

We proposed an improved PSO based on update strategy of double extreme value by analyzing the updating ways of individual extremum and the global extremum influencing the convergence speed and precision of PSO. It improve the individual and global extreme quality and enhance the quality of the population average solution, and accelerate the convergence speed. The results shows that IPSO has good global searching ability, fast convergence speed, solution quality and robust performance optimizations comparing with SGA and PSO by testing with four typical test functions.

Acknowledgements

It is supported by Youth project of National Natural Science fund(41071262), Key Laboratory of intelligent education and information engineering project in Heilongjiang province, Heilongjiang province key discipline of computer application technology(081203).

References

[1] J. Kennedy and R. Eberhart, “Particle Swarm Optimization”, Proc. IEEE Int. Conf. Neural Networks, **(1995)**, pp. 1942-1948.

[2] Y. W. Leung and Y. P. Wang, “An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization”, IEEE Transactions on Evolutionary Computation, vol. 5, no. 1, (**2001**), pp. 41–53.

[3] H. –C. Chen, Y. –W. Chang and R. –C. Hwang, “ A Watermarking Technique based on the Frequency Domain”, Journal of Multimedia, vol. 7, no. 1, (**2012**), pp. 82-89.

[4] Z. Wang, X. Zhao, B. Wan and J. Xie, “Research of BP Neural Network based on Improved Particle Swarm Optimization Algorithm”, Journal of Networks, vol. 8, no. 4, **(2013)**, pp. 947-954.

[5] Y. Shi and R. C. Eberhart, “Parameter Selection in Particle Swarm Optimization”, Proceedings of Evolutionary Programming VII: EP98. New York, **(1998)**, pp. 591–600.

[6] R. C. Eberhart and Y. Shi, "Evolving Artificial Neural Network", Proceedings of Int'1 Conference on Neural Networks and Brain. Beijing, P.R.C., **(1998)**.

[7] Y. Shi and R. C. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization", Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, South Korea, **(2001)**, pp. 101–106.

[8] J. Riget and J. S. Vesterstroem, "A Diversity-Guided Particle Swarm Optimizer–The ARPSO", Technical Report No. 2002–02, Department of Computer Science, University of Aarhus, **(2002)**.

[9] M. Clerc and J. Kennedy, "The Particle Swarm–Explosion, Stability, and Convergence in a Multidimensional Complex Space", IEEE Transactions on Evolutionary Computation, vol. 6, no. 1, **(2007)**, pp. 58–73.

[10] J. Robinson and Y. Rahmat-Samii, "Particle Swarm Optimization in Electronmagnetics", IEEE Transactions on Antennas and Propagation, vol. 52, no. 2, **(2004)**, pp. 397–407.

[11] D. Sharma, A. Chatterjee and A. Rakshit, "A Random Spatial lbest PSO-Based Hybrid Strategy for Designing Adaptive Fuzzy Controllers for a Class of Nonlinear Systems", IEEE Transactions on Instrumentation and Measurement, vol. 61, no. 6, **(2012)**, pp. 1605–1621.

[12] C. –N. Ko, Y. –P Chang and C. –J. Wu, "A PSO Method With Nonlinear Time-Varying Evolution for Optimal Design of Harmonic Filters", IEEE Transactions on Power Systems, vol. 24, no. 1, **(2009)**, pp. 437–444.

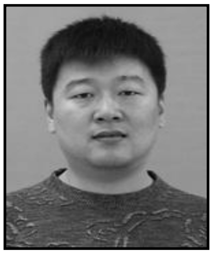
[13] J. G. Vlachogiannis and K. Y. Lee, "Economic Load Dispatch—A Comparative Study on Heuristic Optimization Techniques With an Improved Coordinated Aggregation-Based PSO", IEEE Transactions on Power Systems, vol. 24, no. 2, **(2009)**, pp. 991–1001.

[14] X. Yao, Y. Liu and G. M. Lin, "Evolutionary Programming Made Faster", IEEE Transactions on Evolutionary Computation, vol. 3, no. 2, **(1999)**, pp. 82–102.

[15] Y. W. Leung and Y. P. Wang, "An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization", IEEE Transactions on Evolutionary Computation, vol. 5, no. 1, **(2001)**, pp. 41–53.

[16] S. C. Esquivel and C. A. Coello, "On the Use of Particle Swarm Optimization with Multimodal Functions", In Proceedings of the IEEE Transactions on Evotutionary Computation, vol. 2, no. 12, **(2003)**, pp. 1130-1136.

Authors



Weidong Ji. He received his M.Sc. in Control Theory and Control Engineering (2004), and PhD in Mechanical Design and Theory (2013) from Northeast Forestry University. He is a associate Professor of Computer Science and Information Engineering. Since 2013 he is a Master Instructor. His current research interests include different aspects of Neural Network and Swarm Intelligence.

Jianhua Wang. She received her M.Sc. in Computer Science and Technology (2000) from Harbin Institute of Technology. She is a Professor of Computer Science and Information Engineering. Since 2008 she is dean. Since 2009 she is the laboratory director of "Intelligent Education and Information Engineering" in Heilongjiang Province. Since 2010 she is Doctoral tutor, key academic leader. Her current research interests include different aspects of Artificial Intelligence and Geographic Information Systems.

Jun Zhang. He received his M.Sc. in Computer Science and Technology (2006) from Northeastern University. He is a Professor of Computer Science and Information Engineering. Since 2008 he is Vice-President. Since 2000 he is a Master Instructor, reserve leader of key academic. His current research interests include different aspects of Network Security and Swarm Intelligence.