PARTICLE SWARM OPTIMIZATION BASED ON WEIGHTED AGGREGATION DEGREE AND ADAPTIVE DECISION

Renxia WAN[1], Lijun ZHU[2], Kai LIU[3], Ruidian CHEN[4]

In response to the problem that particle swarm optimization algorithm (PSO) is prone to falling into local optimum and premature convergence in later operations, this paper reconstructs the concept of weighted aggregation based on a redefined similarity to describe the degree of diversity of the population, and adjusts the particle searching space with an adaptive decision to improve the global searching ability of PSO. Optimization ability, convergence speed and stability of the particle swarm algorithm are finally effectively improved. The experimental analysis further shows the effectiveness of the algorithm.

Keywords: particle swarm optimization, similarity, weighted aggregation, adaptive decision

1. Introduction

Particle Swarm Optimization (PSO) was first proposed by Kennedy and Elberhart in 1995[1,2], and began as a simulation of a simplified social system. It is a population-based method inspired from the social behavior of bird flocking or fish schooling, and it has been widely used to solve different kind of optimization tasks. Up to now, many improved PSO algorithms[3,4] have been proposed, and widely used to solve problems in neural networks[5], pattern recognition[6], decision support[7], intrusion detection[8], multi-objective optimization[9].and other academic and applied fields[10-13]. However, PSO may result in premature convergence, local optimum and loss of population diversity[14], in order to overcome these drawbacks, some strategies of increasing the population diversity have been proposed by many scholars, such as solving conflicts between particles[15], introducing speed variation and position variation[16,17], mutation strategy[4], the small probability reinitialization[18], and so on.

Riget and Vesterstorm[19]presented a method for the inverse process of interparticle attraction, they added a mutual exclusion process to the interparticle attraction to enlarge the search space of particles for increasing the diversity of population. Liu and Fan[20] reassigned the moving value of each particle based on similarity and aggregation to improve the global search ability of PSO. This paper describes the diversity of PSO population with the weighted aggregation degree which was produced by a redefined similarity, and adjusts the particle searching space with adaptive decision to improve the global searching ability of PSO.

2. Standard particle swarm optimization

In 1998, Shi and Eberhart [21] introduced the inertia weight ω into the original PSO, and thus produced the standard particle swarm optimization algorithm which was widely accepted by later researchers.

The standard particle swarm optimization algorithm can be described as follows:

Suppose there is a particle swarm of m individuals in a D-dimensional space, the i-th particle is described as $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$ where $i = 1, 2, \cdots, m$. Each position of the particle represents a possible solution, and a fitness value can be calculated by substituting such position into the objective function of the problem. According to the fitness value, PSO determines how

[1] Ningxia Key Laboratory of Intelligent Information and Big Data Processing, North Minzu University, China, e-mail: wanrx1022@126.com
[2] College of Mathematics and Information Science, North Minzu University, China, e-mail: zhulijun1995@sohu.com
[3] Mathematics' Group, Pinghu College, China, e-mail: sddpliukai@163.com
[4] Fujian Hongyang Software co. LTD, China, e-mail: crds002@argox.cn

"good" or "bad" the particle's current position is. Here, we denote the optimal position that has been experienced by X_i as $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$ represents the optimal position of the entire population. The positions and velocities of particles then can be determined as follows:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

where $i = 1, 2, \cdots, m$, $d = 1, 2, \cdots, D$. k is the number of current iteration, r_1, r_2 are the random numbers in the interval $[0,1]$, ω is the inertia weight, c_1 and c_2 are the acceleration factors. v_{id}^{k+1} is the velocity vector.

3. Improved particle swarm algorithm with aggregation

Generally, with the increasing of the number of iterations, all particles will gather to the current optimal position P_g, and individuals' positions are more similar to each other, the PSO algorithm will then converge to the P_g, consequently, such convergence cannot guarantee that PSO finds the real global optimum. In order to overcome the problem that PSO algorithm is prone to lose the population diversity in the later period of operation and falls into the deficiency of local optimization, in this paper, the concept of similarity degree is introduced to measure the relationship between every particle and the current optimal position and the aggregation degree is used to judge particle's density. An adaptive threshold decision is employed to determine the searching space of particle and random variation based on the weighted aggregation is performed to redistribute adaptive value. Thus, the purpose of increasing the diversity of particles is then achieved.

3.1. The concept of aggregation and similarity

Since particle position and velocity in PSO are all described by vectors, we introduce Pearson's correlation coefficient[22] to define the similarity in order to describe the relationship between X_i and P_g.

$$sim = \frac{\operatorname{cov}(X_i, P_g)}{\sqrt{Var(X_i)}\sqrt{Var(P_g)}} \quad (3)$$

Since particles are all the more *sim* is close to 1, the smaller the vector angle of two concerned particles becomes and the more similar these two particles are in a D-dimensional space, then, ; otherwise, the more *sim* is close to 0, the lager the vector angle of two concerned particles becomes and the more dissimilar these two particles are.

In order to reflect the different influences of particles in the swarm, the aggregation degree is used to describe the concentration of the population as the following formula

$$C(k) = \sum_{j=1}^{n} \frac{sim_j}{\sum_{i=1}^{n} sim_i} sim_j \quad (4)$$

where $C(k)$ is the aggregation degree of the *k*-th generation of particle swarm, n is the population size.

The aggregation degree can measure the population concentration base on the similarity between every particle in the swarm and the current optimal position P_g, it adopts a weighting approach, and the more the particle is more similar to P_g, the bigger weightings it may get, then, it accounts for a larger proportion in the calculation of the aggregation degree.

3.2. Adaptive threshold decision

In the standard particle swarm optimization algorithm, Under the guidance of individual optimum and global optimum, the particles are always in the state of contraction, and when an optimal solution (relative to the previous iterations) is found, the searching behaviors of the particles are essentially in stagnation, but this solution may be locally optimal. Riget and Vesterstorm introduced an inverse process of contraction to avoid such situation[19], they changed the coefficient of the two learning factors to a negative form, and got the following formula:

$$v_{id}^{k+1} = \omega v_{id}^{k} - c_1 r_1 (p_{id}^{k} - x_{id}^{k}) - c_2 r_2 (p_{gd}^{k} - x_{id}^{k}) \quad (5)$$

When the population shrinks to a certain state, the diversity falls to a predetermined threshold *(Th*), the algorithm then invokes the formula(5) instead of formula(1) to calculate velocity. And when the diversity of particles returns to be above the predetermined level, the inverse process is then aborted and replaced by the standard PSO processing. But, since the threshold *(Th*) is a predefined constant, it is relatively single for the all iterations and may lead to an excessive searching oscillation in the late iterations.

Referring to the method of controlling inertia weight w with nonlinear function[24], we introduced an adaptive function $\theta(k)$ instead of a constant threshold to control velocity calculation to control the decision of particles' diversities

$$\theta(k) = \left(\frac{k_{\max} - k}{k_{\max}} \right)^{s} \cdot \left(\lambda_{\max} - \lambda_{\min} \right) + \lambda_{\min} \quad (6)$$

where $\lambda_{\min}$, $\lambda_{\max}$ are constants and take values in(0,1), $\lambda_{\min} \le \lambda_{\max}$. $k_{\max}$ is the maximum number of iterations, *k* is the number of current iteration. *s* is an exponential coefficient. We set *s*=1 in our experiments, and our experimental results also show that $\lambda_{\max}$ =0.9, $\lambda_{\min}$ =0.4 is the best choice in our most instances.

The individual's velocity is then decided by the strategy as

$$\begin{cases} If\ C(k) < \theta(k) \cdot Th, \text{ update } v_{id}^{k} \text{ with the formula (1)} \\ If\ C(k) > \theta(k) \cdot Th, \text{ update } v_{id}^{k} \text{ with the formula (5)} \end{cases}$$

3.3 Updating particles' positions with aggregation

A particle swarm optimization algorithm with similarity was proposed by Liu and Fan in 2007[20], it embodies the idea that when the population concentration reaches a certain degree, the particles' positions turn to update, and the diversity of individual then increases. Here, we make use of this idea with some improvements

If $rand < h \cdot C(k) \cdot sim$

$$\begin{cases} x_{id_0} = (x_{id\max} - x_{id\min}) \cdot p_{gd} \cdot rand \cdot Gausssian(\mu, \sigma^2), & d_0 = U[1, D] \\ x_{id} = p_{gd} \,, & d \in U[1, D] \ and \ d \ne d_0 \end{cases} \quad (7)$$

where $rand$ is the random number in the range $[0,1]$, $x_{id\max}$ and $x_{id\min}$ are the upper and lower limits of the search space, $Gaussian(\mu, \sigma^2)$ is the Gaussian function, $\sigma = (\sigma_{\max} - \sigma_{\min}) \cdot \frac{t}{T}$, and in this paper, we set $\sigma_{\max}$ =1.0, $\sigma_{\min}$ =0.1. h is a constant parameter, the bigger h is , the more probabilistic mutation operation is, PSO is then apt to jump out of local optimum, in this paper, we refer to literature[14] and set *h*=3

3.4 The CSPSO algorithm

According to the solutions described above, an adaptive PSO based on weighted aggregation degree and adaptive decision is then proposed in this paper. The logical flow of the new PSO algorithm called as CSPSO is presented as follows:

Step 1 Create and initialize with a group of random particles and set parameters.

Step 2 Evaluate each individual's fitness.

Step 3 Determine the global best position P_g and the current individual best position P_i .

Step 4 Calculate the sim from the formula (3) and the $C(k)$ from the formula (4).

Step 5 Update the individual's velocity.

$$\begin{cases} If\ C(k) < \theta(k)\cdot Th, \text{ then update the velocity by the formula (1)} \\ If\ C(k) > \theta(k)\cdot Th, \text{ then update the velocity by the formula (5)} \end{cases}$$

Step 6 Update the individual's position.

If $rand < h\cdot C(k)\cdot sim$

$$\begin{cases} x_{id} = (x_{id\max} - x_{id\min})\cdot p_{gd}\cdot rand\cdot Gaussian(\mu,\sigma^2), & d = U[1,D] \\ x_{id} = p_{pd} & else \end{cases}$$

Step 7 If requirements are met, then stop. Otherwise, go back to step 2

4. Experimental results and performance comparison

Some typical test functions were chosen to evaluate the performance of the algorithm.

4.1 Test functions

The typical test functions listed in Table1 are used are widely used in performance comparison of global optimization algorithms. All the test functions are divided into two groups based on their significant physical properties and shapes. The first group consists of five unimodal functions *(f*1-*f*4) where there is only one mode (global optimum) in its geometric distribution of each function. The second group includes four multimodal functions (f_5-f_9) where every function has multiple local minima, and finding the global optimum is the typically challenging work.

Table 1

Test functions used in experiments

f	Name	Function	Optimum (x)	Minimum f(x)
f_1	Sphere	$f_1(x)=\sum_{i=1}^{n} x_i^2 \quad \lvert x_i\rvert \le 100$	$(0,\ldots,0)$	0
f_2	Schwefel 1.2	$f_2(x)=\sum_{i=1}^{n}(\sum_{j}^{i} x_j)^2 \quad \lvert x_i\rvert \le 100$	$(0,\ldots,0)$	0
f_3	Schwefel 2.21	$f_3(x)=\max_{i=1}^{n}\{\lvert x_i\rvert\} \quad \lvert x_i\rvert \le 100$	$(0,\ldots,0)$	0
f_4	Step	$f_4(x)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2 \quad \lvert x_i\rvert \le 100$	$(0,\ldots,0)$	0
f_5	Rastrigin	$f_5(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10) \quad \lvert x_i\rvert \le 5.12$	$(0,\ldots,0)$	0

f_6	Girewank	$f_6(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1 \quad \lvert x_i \rvert \le 600$	$(0,L\ ,0)$	0
f_7	Generalized Penalized	$f_7(x)=\frac{\pi}{n}\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+\frac{\pi}{n}(y_n\text{-}1)$ $+\frac{\pi}{n}[10\sin^2(\pi y_i)]+\sum_{i=1}^{n}u(x_i,10,100,4) \quad \lvert x_i \rvert \le 50$	$(1,L\ ,1)$	0
f_8	Kowalik	$f_8(x)=\sum_{i=1}^{11}[a_i-\frac{x_i(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}]^2 \quad \lvert x_i \rvert \le 5$	(0.1928, 0.1928, 0.1231, 0.1358)	3.075×10^{-4}
f_9	Hartman	$f_9(x)=-\sum_{i=1}^{4}c_i\exp[-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2] \quad 0\le x_j \le$	(0.201, 0.15, 0.477, 0.275, 0.311, 0.657)	-3.32

4.2 Parameter settings

In order to measure the optimization ability, convergence speed and stability of algorithm, CSPSO and other three available ECTs were evaluate by the nine unimodal and multimodal functions given in Table 1. Population size in these algorithms is set at 40, dimension is set at 30 (Kowalik function is set at 4, Hartman function is set at 6), the acceleration factors c_1 and c_2 are all set at 1.49445, the number of iterations $K_{\max}$ is set at 500, the initial threshold *Th*=5. We adopted a nonlinear function as the inertia weight: $\omega=0.9-0.5\times(k/k_{\max})^2$ [24], All experiments were repeated 30 times independently, namely, N_{run}=30.

In order to evaluate performance of CSPSO and its matching algorithms in terms of accuracy, and reliability, BFV (Best Fitness Value: defined as the minimum optimized *f*(*x*) value obtained from N_{run} independent runs), WFV (Worst Fitness Value: defined as the maximum optimized *f*(*x*) value obtained from Nrun independent runs), MFV (Mean Fitness Value: defined as the average of the Nrun BFVs) and σ_0^2 (Variance of the Nrun BFVs) are defined as the performance measures.

Experimental results are shown in Table 2 and Fig. 1.

Table 2

Performance comparison between CSPSO and its competitors on test functions

Function	Algorithm	Fitness Value（FV）			σ_0^2
		WFV	*BFV*	*MFV*	
f_1	PSO	13.0606	2.2331	7.3282	10.4400
	ARPSO	22.2565	5.1831	10.8257	30.7887
	SPSO	18.7574	3.0807	7.8425	17.4451
	CSPSO	6.6628	1.6580	3.1689	2.4152
f_2	PSO	14.0942	4.9555	9.3483	10.7214
	ARPSO	24.2667	11.0227	16.8064	20.3572
	SPSO	33.2447	8.8509	21.6020	52.1040
	CSPSO	4.5907	0	1.9486	3.0339
f_3	PSO	2.0952	1.1746	1.8002	0.1184
	ARPSO	2.4113	1.5132	1.8339	0.0823
	SPSO	2.3620	1.7274	2.0780	0.0642

	CSPSO	1.1768	0.5869	0.9466	0.0311
f_4	PSO	1992	706	1219.4545	155555.272
	ARPSO	2313	742	1483.4545	264085.672
	SPSO	2264	519	1306.0909	312116.090
	CSPSO	23	1	9.8182	68.1636
f_5	PSO	226.1740	92.1761	126.9195	1670.5657
	ARPSO	276.3192	130.6707	168.7579	1748.4357
	SPSO	171.5772	112.3695	144.2489	366.6862
	CSPSO	94.3342	33.2073	57.5121	347.1176
f_6	PSO	0.6430	0.1596	0.3464	0.0154
	ARPSO	0.6081	0.2716	0.4915	0.0147
	SPSO	0.4381	0.1752	0.2605	0.0066
	CSPSO	0.1622	0	0.0791	0.0028
f_7	PSO	1.3299	0.2595	0.5490	0.0850
	ARPSO	1.8966	0.3185	0.8555	0.2175
	SPSO	1.2106	0.1303	0.8044	0.1798
	CSPSO	0.1673	0.0260	0.0828	0.0024
f_8	PSO	0.001	0.0004	0.0009	3.0667e-08
	ARPSO	0.0018	0.0007	0.0012	1.1156e-07
	SPSO	0.0011	0.0003	0.0007	1.0267e-07
	CSPSO	0.0009	0.0003	0.0004	2.0351e-09
f_9	PSO	-3.2031	-3.3220	-3.2923	0.0029
	ARPSO	-1.5781	-3.3154	-2.9187	0.3818
	SPSO	-3.2031	-3.3220	-3.2626	0.0039
	CSPSO	-3.2031	-3.3220	-3.3022	0.0021

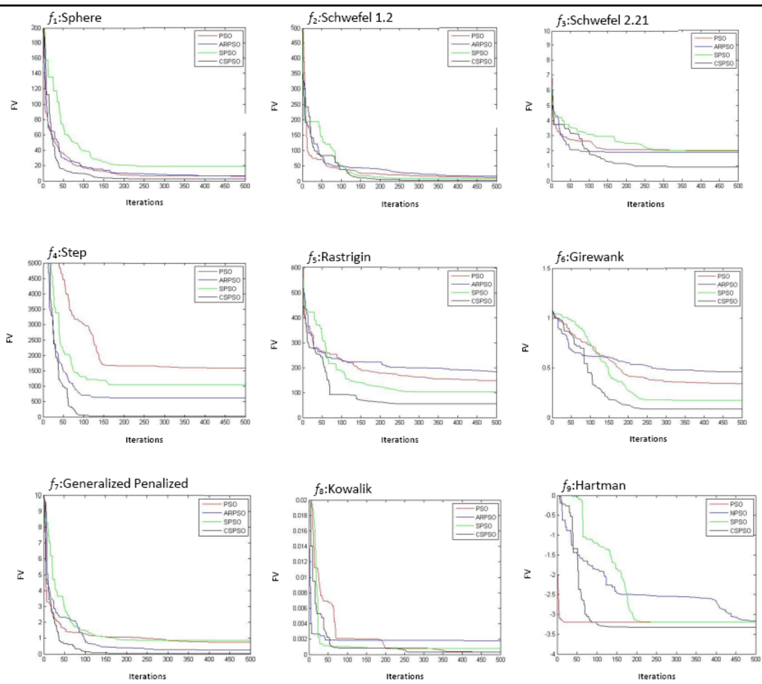

Fig. 1. Comparison of FV between CSOPS and three ECTs

4.3 Observation and analysis

Table 2 shows that CSPSO has the best MFVs in all competitive algorithms over the given test functions, it indicates that the proposed CSPSO performs better than other three ECTs in the term of global optimizing accuracy and effectiveness. Table2 also shows that the variance σ_0^2 of CSPSO is always smallest in all competitive algorithms, meaning that the proposed CSPSO outperforms other three ECTs in the term of global optimizing reliability and stability.

When the number of iterations is large enough, some PSO algorithms can achieve a certainly satisfactory optimal position. In our experiment, we limited the number of iterations to 500 to examine the convergence and time efficiency of the algorithm in a relatively short time. Figure 1 is the effect curves of one experiment randomly selected from our 30 experiments. From Figure 1, we can see that CSPSO can achieve much better convergence accuracy than other three ECTs in limited iterations, and CSPSO is faster in finding out the approximate optimal solution than other three ECTs. It means that the proposed CSPSO outperforms other three ECTs in the term of success rate and execution time of global optimizing.

5. Conclusions

A novel PSO algorithm (CSPSO) was proposed whose performance is found to be superior to its ECTs. CSPSO controls the diversity of the population with the weighted aggregation and adopts the adaptive threshold decision to improve the global searching ability of individual. From the experimental analysis, we have shown that CSPSO outperforms its ECTs in terms of convergence, accuracy, reliability and executing time.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61662001, 61762002), Young and Middle-aged Talents Training Program of National Ethnic Affair Commission (2016GQR06). Ningxia First-class Construction Discipline Program (NXYLXK2017B09).

REFERENCES

[1] *J. Kennedy, R. C. Eberhart*, Particle swarm optimization, In: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, (1995), 1942-1948.

[2] *R. C. Eberhart, J. Kennedy*, A new optimizer using particle swarm theory, In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Piscataway, NJ, USA, (1995), 39–43.

[3] *L. T. Al-Bahrani, J. C. Patra*, A novel orthogonal PSO algorithm based on orthogonal diagonalization, Swarm and Evolutionary Computation, 40(2018), 1-23.

[4] *A. R. Jordehi*, Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimization problems, Applied Soft Computing Journal, 26(2015), 401-417.

[5] *J. Wang, T. Kumbasar*, Parameter optimization of interval Type-2 fuzzy neural networks based on PSO and BBBC methods, IEEE/CAA Journal of Automatica Sinica, 6(2019), 247-257.

[6] *D. Dang, R. Guibadj, A. Moukrim*, An effective PSO-inspired algorithm for the team orienteering problem, European Journal of Operational Research, 229(2013), 332-344.

[7] *F. Azevedo, Z. A. Vale, P. B. D. M. Oliveira*, A decision-support system based on particle swarm optimization for multiperiod hedging in electricity markets, IEEE Transactions on Power Systems, 22(2007), 995-1003.

[8] *S. Jamali, G. Shaker*, SO-SFDD: Defense against SYN flooding DoS attacks by employing PSO algorithm, Computers and Mathematics with Applications, 63(2012), 214-221.

[9] *T. Niknam, H. Z. Meymand, H. D. Mojarrad,* A practical multi-objective PSO algorithm for optimal operation management of distribution network with regard to fuel cell power plants, Renewable Energy, 36(2011), 1529-1543.

[10] *A. Kornelakis, Y. Marinakis*, Contribution for optimal sizing of grid-connected PV-systems using PSO, Renewable Energy, 35(2010), 1333-1341.

[11] *D. Dang, R. N. Guibadj, A. Moukrim*, An effective PSO-inspired algorithm for the team orienteering problem, European Journal of Operational Research, 229(2013), 332-344.

[12] *W. Zhang, J. Liu, L. Fan, Y. Liu, D. Ma*, Control strategy PSO, Applied Soft Computing, 38(2016), 75-86.

[13] *Y. Soufl, M. Bechouat, S. Kahla*, Fuzzy-PSO controller design for maximum power point tracking in photovoltaic system, International Journal of Hydrogen Energy, 42(2017), 8680-8688.

[14] *F. Marini, B. Walczak*, Particle swarm optimization (PSO): A tutorial, Chemometrics and Intelligent Laboratory Systems, 149(2015), 153-165.

[15] *T. Krink, J. S. Vesterstrom, J. Riget*, Particle swarm optimization with spatial particle extension, In: Proceedings of the Congress on Evolutionary Computation, Honolulu, Hawaii, USA, (2002), 1474-1497.

[16] *X. F. Xie, W. J. Zhang, Z. L. Yang,* A dissipative particle swarm optimization, In: Proceedings of the IEEE International Conference on Evolutionary Computation, Honolulu, Hawaii, USA, (2002), 1666 -1670.

[17] *P. Liu, J. Liu*, Multi-leader PSO (MLPSO): A new PSO variant for solving global optimization problems, Applied Soft Computing, 61(2017), 256-263.

[18] *F. V. D. Bergh*, An Analysis of Particle Swarm Optimizers, Ph.D. Dissertation, University of Pretoria, Pretoria, South Africa, 2002.

[19] *J. Riget, J. S. Vesterstroem*, A Diversity-guided Particle Swarm Optimizer-the ARPSO, Technical report, University of Aarhus, Aarhus, Denmark, 2002.

[20] *J. H. Liu, X. P. Fan*, A new particle swarm optimization algorithm based on similarity, Control and Decision, 20(2008), 1154-1159.

[21] *Y. Shi, R. Eberhart*, A modified particle swarm optimizer. In: IEEE Word Congress on Computational Intelligence, Anchorage, Alaska, USA, (1998), 69-73.

[22] *J. L. Rodgers, W. A. Nicewander*, Thirteen ways to look at the correlation coefficient, The American Statistician, 42 (1988), 59-66.

[23] *A. Chatterjee, P. Siarry*, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Computers and Operations Research, 33(2006), 859-871.

[24] *L. Li, P. Niu,* Particle Swarm Optimization Algorithm, Metallurgy Industry Press, Beking, China, 2009.