Chinese Journal of Electronics
Vol.22, No.1, Jan. 2013

A Hybrid Simplex Search and Modified Bare-bones Particle Swarm Optimization*

WANG Panpan, SHI Liping, ZHANG Yong and HAN Li

(*School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221008, China*)

Abstract — In order to enhance global convergence capability of particle swarm optimization, this paper proposes a novel hybrid algorithm, called SM-MBBPSO, based on the Nelder-Mead Simplex method (SM) and a Modified bare-bones particle swarm optimization (MBBPSO). In this algorithm, a new strategy based on K-means clustering is proposed to combine the powerful global search capability of MBBPSO and the high accurate local search capability of SM. This makes the proposed algorithm achieve a nice balance between exploitation and exploration capability. Meanwhile, an adaptive reinitialization strategy on inactive particles is proposed to help the swarm get away from local optimal positions. Finally, simulation results on benchmark functions demonstrate the effectiveness of the proposed algorithm.

Key words — Bare-bones particle swarm optimization, Nelder-Mead simplex method, Inactive particle, K-means clustering.

I. Introduction

Particle swarm optimization (PSO) which searches for the solution space by exchanging experience among particles in a swarm, was firstly proposed by Kennedy and Eberhart in 1995[1]. Compared with other optimization algorithms, PSO has many advantages such as simplicity, fast convergence and population-based feature, and has been paid widely attention to scientific research and engineering practice[2−6]. However, the traditional PSO requires users to tune such control parameter as inertia weight and acceleration coefficients in order to find desirable solutions for the optimized problem. To overcome this disadvantage, many improved methods have been proposed. In the Ref.[7], Bare-bones particle swarm optimization (BBPSO) which doesn't need to set the inertia weight and the acceleration coefficients was proposed by Kennedy in 2003. In the BBPSO, the position of each particle is randomly selected from the Gaussian distribution based on two leaders, *i.e.*, the global best position and the personal best position of particle.

However, like many standard PSOs, BBPSO is prone to premature convergence, as a result of the decrease of population diversity with number of iteration. In order to improve the efficiency of BBPSO, a number of modified BBPSO algorithms have been proposed recent years. Omran *et al.*[8] proposed an effective algorithm by combining concepts of BBPSO and the recombination operator of different evolution. Krohling and Mendel[9] introduced the jump approach into BBPSO algorithm. Zhang *et al.*[10] proposed a BBPSO with mutation and crossover operations. Although these algorithms are able to improve the convergence of BBPSO, they cannot present an excellent tradeoff between exploitation and exploration.

In order to enhance the search capability of BBPSO, a new hybrid approach based on K-means clustering is proposed in this paper by combining a Modified bare-bones particle swarm optimization (MBBPSO) and the Nelder-Mead Simplex method (SM). Moreover, an adaptive reinitialization strategy on inactive particles is proposed to maintain diversity of swarm and improve search efficiency of particles. Lastly, the proposed algorithm is validated by using several benchmark functions and compared against three highly competitive algorithms. Results indicate the superiority of the proposed scheme.

II. Basic Concepts

1. Particle swarm optimization

Particle swarm optimization (PSO) is a stochastic, population-based optimization approach modeled after the simulation of social behavior of bird flocks. In a PSO system, a swarm of individuals (called particles) fly through a D-dimensional search space. Each particle represents a candidate solution for the problem and adjusts its position by tracking two optima. One is its personal best position p_i, namely, the best position found by it so far. The other is the global best position p_g, namely, the best position found by its neighborhood particles so far. In each generation, the velocity $\boldsymbol{v}_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,D})$ and position $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$ of particle are updated by the following formulas[1], where D is the total dimension number of search space.

$$\begin{aligned} v_{i,j}(t+1) =& w v_{i,j}(t) + c_1 r_1 (p_{i,j}(t) - x_{i,j}(t)) \\ &+ c_2 r_2 (p_{g,j}(t) - x_{i,j}(t)) \qquad (1) \\ x_{i,j}(t+1) =& x_{i,j}(t) + v_{i,j}(t+1) \qquad (2) \end{aligned}$$

where w is an inertia weight to control exploration in the search space, c_1 and c_2 are two positive constants, r_1 and r_2 are two

*Manuscript Received Dec. 2011; Accepted Apr. 2012. This work is supported by the National Natural Science Foundation of China (No.61005089), and the Scientific and Technological Research Foundation of China University of Mining and Technology (No.2008A020).

random numbers within [0,1]. $v_{i,j} \in [-v_{\max}, v_{\max}]$, where $v_{\max}$ is set by the user.

Clerc and Kennedy proved that each particle converges to a weighted average of its personal best and neighborhood best position[11], that is,

$$\lim_{t\to\infty} x_{i,j}(t) = \frac{c_1 p_{i,j} + c_2 p_{g,j}}{c_1 + c_2}. \tag{3}$$

This theoretically derived behavior provides support for the Bare-bones particle swarm optimization (BBPSO) developed by Kennedy[7]. In BBPSO, the formula for updating particle's velocity is eliminated, and a Gaussian sampling based on the p_i and p_g information is used as follows.

$$x_{i,j}(t+1) = N(\mu_{i,j}(t), \sigma_{i,j}^2(t)) \tag{4}$$

where $\mu_{i,j}(t) = (p_{i,j}(t) + p_{g,j}(t))/2$ and $\sigma_{i,j}^2(t) = |p_{i,j}(t) - p_{g,j}(t)|$ are the mean and standard deviation of Gaussian distribution, respectively. Compared to the canonical PSO, BBPSO is obviously control parameter-free and more compact. So it is natural to apply it to some real application problems.

2. The Nelder-Mead simplex search method

The Nelder-Mead simplex Search method (SM) is a traditional, deterministic and local search method, which is proposed by Nelder and Mead (1965)[12]. The operations of this method rescale the simplex based on the local behavior of the function by using four basic procedures: reflection, expansion, contraction and shrinkage. Through these procedures, the simplex can successfully improve itself and get closer to the optimum. Since this method is easy to use and does not need gradient information of the function under exploration, it has been widely used in unconstrained optimization. However, the SM method is very sensitive to the choice of initial points and not guaranteed to attain the global optimum.

III. The Proposed Approach

As we know, the simplex search method is straightforward in an algorithmic sense and computationally efficient. However, since the SM only uses local information, there is no guarantee that the global optimum is found finally. In contrast, since the BBPSO explores the global search space with global information of swarm, it is less likely to be trapped in local optima. However, it needs higher computational cost to locate the optimal solution at the end of algorithm. In short, the BBPSO method shows good exploration capability and the simplex method has good exploitation capability. In order to get a balance between exploitation and exploration, it is reasonable to design a hybrid method by making most use of the advantage of each method.

1. The modified BBPSO method

Despite the simplicity and efficiency of BBPSO algorithm, it still suffers from premature convergence[7]. To resolve this problem, this section proposes a Modified bare-bones particle swarm optimization (MBBPSO), by designing an adaptive reinitialization strategy of inactive particle.

With the increasing number of iterations, in BBPSO the particles will be drawn towards the global best particle, and the swarm diversity will decrease significantly. From Eq.(4), we can see that the smaller the distance between the p_i and the p_g is, the less the probability that the particle moves to a new position is. Herein, if the global best particle is a local optimum of the optimized problem, the small update probability of position will make swarm trap into local optimum. Due to less contribution on evolution, we define these particles whose fitness are difficult to improve as inactive particles.

Definition 1 For a particle, if its fitness is no improvement for η cumulative iterations, we will consider it to be an inactive particle, where η is inactive coefficient.

To enhance the search efficiency of inactive particle and help BBPSO escape from local optimum, a strategy of adaptive reinitialization is proposed. To report the change of particles, a *stagnation_iteration* is used to monitor the fitness value for each particle. At each iteration, if there is no fitness improvement for a particle, then the *stagnation_iteration* will be increased by one, or else, decreased by one. If *stagnation_iteration* $\geq \eta$, we will consider this particle to be an inactive particle, and then reinitialize this particle's position. Meanwhile, the inactive particle will forget its memory and select current position to be the personal best position. This dynamic reinitialization operator will help inactive particle escape from local optimum and improve its search efficiency. In order to maintain diversity of swarm and reduce the possibility of local optimum during the whole process, the inactive coefficient η will be updated by the following equation.

$$\eta = \lfloor \eta_{\min} + (\eta_{\max} - \eta_{\min}) * \mathrm{e}^{-\varepsilon * t/t_{\max}} \rfloor \tag{5}$$

where $t_{\max}$ is maximum number of iterations, t is current number of iterations, ε is set to control the decreasing speed of η, $\lfloor \bullet \rfloor$ indicates rounding down.

The key steps of MBBPSO are described as follows:

Step 1 Initialization. Initialize particles, their personal best positions, global best position and *stagnation_iteration*, set the parameters such as size of swarm and maximum generation.

Step 2 Calculate each particle's fitness.

Step 3 Update the personal best position of inactive particles p_q, and the personal best position of the rest particles p_i, according to Eq.(6).

$$\begin{aligned} p_q(t+1) &= x_q(t+1) \\ p_i(t+1) &= \begin{cases} p_i(t), & f(p_i(t)) \leq f(x_i(t+1)) \\ x_i(t+1), & f(p_i(t)) > f(x_i(t+1)) \end{cases} \quad i \neq q \end{aligned} \tag{6}$$

Step 4 Reinitialize *stagnation_iteration*$[q]$ of inactive particles q. For the rest particles i, if their fitness do not change, the *stagnation_iteration*$[i]$ will be increased by 1, or else, decreased by 1.

Step 5 Update the global best position p_g.

Step 6 Update position of each particle. If the particle is an inactive particle, reinitialize its position at random; otherwise, update position according to Eq.(4).

Step 7 If the termination criteria is satisfied, stop the iteration procedure; otherwise, go to Step 2.

2. Hybrid SM-MBBPSO method

In this section, the proposed hybrid algorithm (SM-MBBPSO) is introduced. In order to combine effectively the powerful global search capability of MBBPSO and the high accurate local search capability of SM, a new combination strategy based on K-means clustering is proposed.

To improve the performance of PSO, the SM is introduced in many different ways. Hsu and Gao[13] applied SM to the global best particle every k iterations. Although this way increase the convergence speed of PSO, they could not make use of other particles' experience. Wang and Qiu[14] applied SM to all particles every several iterations. Although this way makes use of all particles' experience, the computational cost is higher. To overcome the above problems, this section proposes a new combination way based on K-means clustering to incorporate SM into MBBPSO. In the process of MBBPSO, all the particles are classified into k_c cluster first, and the SM is only applied to the center particle of each cluster every K iterations. This new combination strategy not only makes use of the whole swarm information effectively, but also has a low computational cost. In addition, in order to increase the quality of solution further, SM is also applied to the global best particle at every iteration.

Based on K-means clustering algorithm, the key steps of SM-MBBPSO are outlined as follows.

Step 1 Initialization. Initialize particles, their personal best positions, global best position and *stagnation_iteration*, set the parameters such as size of swarm, maximum generation, the number of cluster k_c, interval generation K and SM parameters.

Step 2 Calculate each particle's fitness.

Step 3 Update p_q and p_i according to Eq.(6).

Step 4 Reinitialize *stagnation_iteration*$[q]$ of inactive particles q. For the rest particles i, if their fitness do not change, the *stagnation_iteration*$[i]$ will be increased by 1, or else, decreased by 1.

Step 5 If $\frac{t}{K} = \lfloor \frac{t}{K} \rfloor$, perform the Nelder-Mead simplex search method.

(1) Classify particles into k_c clusters. And select center particle from every cluster.

(2) Apply SM to the center particles.

Step 6 Update the global best position p_g.

Step 7 Apply SM to the global best particle.

Step 8 Update each particle position. If the particle is an inactive particle, reinitialize its position at random; otherwise, update the position of particle according to Eq.(4).

Step 9 If the termination criteria are satisfied, stop the iteration procedure; otherwise, go to Step 2.

IV. Comparison of Results

To analyze the convergence speed and global convergence of the hybrid SM-MBBPSO method, we compare the SM-MBBPSO with some existing modified BBPSOs for four benchmark functions in this section.

1. Test functions

In order to test the performance of proposed algorithm, four benchmark functions Sphere, Rosenbrock, Schwefel's problem 2.26[10] and Griewank are selected. Many researchers[8−10] have applied these test problems to examine their proposed algorithms. The detail of these test problems can be obtained from the related literatures and Table 1.

2. Compared algorithms and parameter settings

In this comparison, three modified BBPSO algorithms, *i.e.*, DE-BBPSO, BBPSO+CJ, BBPSO+MC, are selected. Where the DE-BBPSO[8] is a hybrid of the BBPSO and differential evolution, the BBPSO+CJ[9] combines the BBPSO with a jump strategy, the BBPSO+MC[10] incorporates mutation and crossover operators of differential evolution into BBPSO.

Each experiment is terminated only when the maximum generation (500) has been reached. The stopping criterion of the simplex search method used in SM-MBBPSO is based on the simplex size. This stopping criterion proposed by Dennis and Woods[15], is defined as:

$$\frac{\max_i \|x_i - x_{n+1}\|}{\Delta} \leq v, \quad \Delta = \max(1, \|x_{n+1}\|) \qquad (7)$$

where $i = 1, 2, \cdots, n$, n is the total dimension number of x_i, $v = 1 \times 10^{-4}$, $\|\cdot\|$ denotes the Euclidean norm. The simplex search method will stop when either Eq.(7) is satisfied or the number of iterations reaches the SM maximum generation. In the SM-MBBPSO algorithm, the SM parameters are important. According to the Ref.[12], the standard, nearly universal choices for these parameter values are $\alpha = 1$, $\gamma = 2$, $\beta = 0.5$, and $\delta = 0.5$. We also use these values in the hybrid algorithm. The parameter configurations for all the four algorithms are summarized in Table 2.

3. Simulation results and analysis

In order to validate the robustness of the proposed algorithm, each function was solved 50 times, and different random number seed was used at each time. The performance of the proposed algorithm is compared with other algorithms based on Best value (BV), Mean value (MV), Worst value (WV), Success rate of optimization (SR) and running Time (T). All

Table 1. Test functions

Description of test functions	Dimension (D)	Search space (S)	Initialization range (I)	Global minimum position ($x_{\min}$)	Global minimum ($f_{\min}$)
Sphere $f_1(x) = \sum_{i=1}^{D} x_i^2$	30	$(-100, 100)^D$	$(-100, 100)^D$	$x_i = 0$	0
Rosenbrock $f_2(x) = \sum_{i=1}^{D-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	30	$(-30, 30)^D$	$(-30, 30)^D$	$x_i = 1$	0
Schwefel's problem 2.26[10] $f_3(x) = -\sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	30	$(-500, 500)^D$	$(-500, 250)^D$	$x_i = 420.9687$	−12569.487
Griewank $f_4(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$(-600, 600)^D$	$(300, 600)^D$	$x_i = 0$	0

Table 2. Parameter configurations for the selected algorithms

Algorithms	Population size	Maximum generation	Other parameters
BBPSO+CJ	50	500	With Cauchy jump; $maximum\_stagnation\_interval = 5$; Scaling parameter: 1.1, 1.1, 20, 1.1 for each function respectively.
DE-BBPSO	50	500	Differential evolution probability of reproduction: 0.7.
BBPSO-MC	50	500	—
SM-MBBPSO	20	500	Maximum generation of SM applied to global best particle: $smj\_t = 350$; Maximum generation of SM applied to cluster center particles: $smc\_t = 50$; $k\_c = 4$; $K = 5$; $\eta_{\min} = 5$; $\eta_{\max} = 50$; $\alpha = 1$; $\gamma = 2$; $\beta = 0.5$; $\delta = 0.5$.

Table 3. Experimental results for test functions

Functions	Algorithms	BV	MV	WV	SR	T/s
$f_1(x)$	BBPSO+CJ	4.373 e-06	8.138 e-05	3.074 e-04	**1**	16.0
	DE-BBPSO	**2.141 e-18**	6.188 e-08	2.871e-06	**1**	7.4
	BBPSO-MC	9.529 e-06	1.209 e-04	3.192 e-04	**1**	**5.3**
	SM-MBBPSO	3.481 e-13	**2.207 e-12**	**9.975e-12**	**1**	13.5
$f_2(x)$	BBPSO+CJ	17.059 023	111.014 293	509.493 802	0	14.6
	DE-BBPSO	28.448 691	153.608 671	1 012.475 689	0	7.0
	BBPSO-MC	13.389 214	79.651 461	158.398 867	0	**5.2**
	SM-MBBPSO	**0**	**2.359 e-10**	**3.899 e-09**	**1**	13.0
$f_3(x)$	BBPSO+CJ	−11 564.346 110	−10 787.461 189	−9 351.708 850	0	14.8
	DE-BBPSO	−11 503.541 607	−10 540.907 978	−9 075.401 045	0	7.1
	BBPSO-MC	−12 073.150 627	−11 614.036 071	−10 748.914 190	0	**5.3**
	SM-MBBPSO	**−12 569.486 618**	**−12 569.486 618**	**−12 569.486 618**	**1**	12.1
$f_4(x)$	BBPSO+CJ	2.409 e-05	1.666 e-02	9.189 e-02	0.26	22.4
	DE-BBPSO	3.333 e-011	7.318 e-01	12.187 814	0.14	**7.2**
	BBPSO-MC	1.450 e-04	7.422 e-03	5.084 e-02	0.34	9.2
	SM-MBBPSO	**2.232 e-14**	**1.075 e-12**	**3.609 e-11**	**1**	15.7

the results of metric are averaged over 50 runs for each benchmark function. In all experiments, a run is considered to be successful if its result is close to the known global optimum within 0.001. The results of experiment are showed in Table 3 and Fig.1. The statistically significant best metric values have been shown in bold.

For the simple test problems f_1, all the algorithms could find the global optimum with 100% successful rate. However, the SM-MBBPSO significantly outperforms BBPSO+CJ and BBPSO-MC, in terms of BV, MV, WV and SR. Although DE-BBPSO sometimes could get a better solution and had lower computation cost than SM-MBBPSO, its solution quality is very unstable. From Fig.1(a), we can see that SM-MBBPSO converged to the near-optimal solution faster than the other three algorithms on this function. The Rosenbrock function f_2 is known as banana function and is a difficult task to achieve its global optimum. The Rosenbrock function is proved to be hard for all the algorithms except SM-MBBPSO, as Table 3 and Fig.1(b) showed. On this function, only SM-MBBPSO could find the global optimum with 100% successful rate. This owes mainly to the right hybrid way between SM and MBBPSO, $i.e.$, the approach based on K-means clustering.

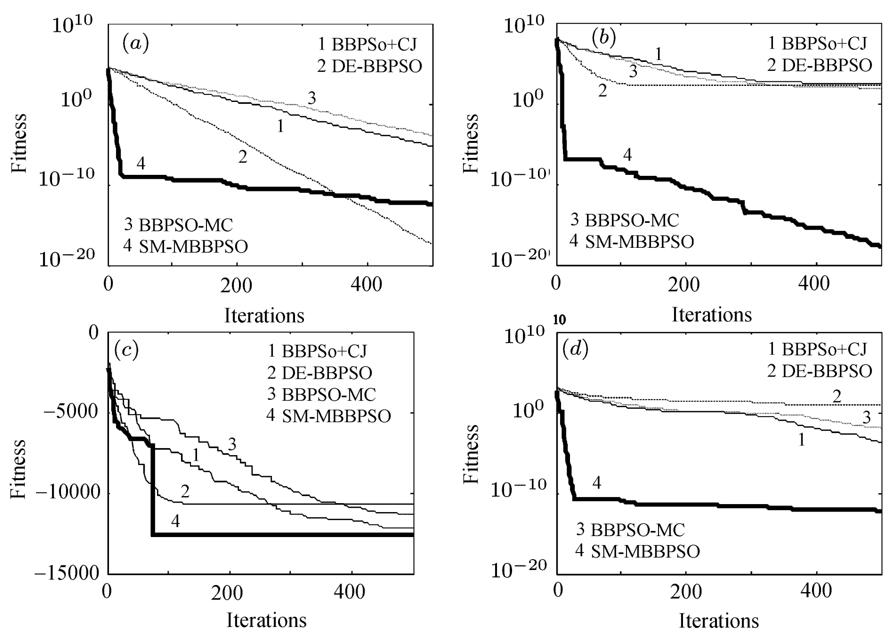

Fig. 1. Performance comparison of the different methods on test functions. (a) Sphere; (b) Rosenbrock; (c) Schwefel's problem 2.26; (d) Griewank

For the Schwefel's problem 2.26 function f_3, its surface is composed of a great number of peaks and valleys. The search algorithms are potentially prone to convergence in the wrong direction when optimizing this function. From Table 3, it is clear that SM-MBBPSO was the winner which was capable of converging to the global optimum with 100% successful rate, whereas the other three algorithms were not able to find the global optimum at all. It is important to point out that, the adaptive reinitialization procedure of MBBPSO and the hybrid way based on K-means clustering played a critical role to help the proposed algorithm to jump out of local optima. With regard to test functions f_4, SM-MBBPSO shows its great search capability and the prominent capability of escaping from local optima. Only the SM-MBBPSO could find the global optimum with 100% successful rate. From Table 3 and Fig.1(d), it can be observed that the performance of SM-MBBPSO is the best except running time.

V. Conclusions and Future Work

In this paper, we developed a novel hybrid optimization method, called hybrid SM-MBBPSO algorithm, by using a new hybrid strategy, *i.e.*, K-means clustering algorithm. The exploration capability of MBBPSO and the exploitation capability of SM make the SM-MBBPSO show strong capability of global convergence and fast convergence speed. Moreover, an adaptive reinitialization of inactive particles is proposed to enhance the global search capability of BBPSO.

Compared with several algorithms, the simulation results demonstrate that SM-MBBPSO is well suited to those complex unimodal or multimodal functions with high dimension. As a consequence, the SM-MBBPSO may be a promising and viable tool to deal with complex engineering optimization problems. In future, it is desirable to further apply SM-MBBPSO to solving more complex real-world optimization problems. This will allow us to have a more pragmatic assessment of the usefulness of SM-MBBPSO.

References

[1] J. Kennedy, R.C. Eberhart, "Particle swarm optimization", *Proc. of IEEE International Conference on Neural Networks*, Piscataway, New Jersey, pp.1942–1948, 1995.

[2] J. van Ast, R. Babuška, B. De Schutter, "Particle swarms in optimization and control", *Proc. of 17th World Congress International Federation of Automatic Control*, Seoul, Korea, pp.5131–5136, 2008.

[3] Q. Lü, S.R. Liu, "A particle swarm optimization algorithm with fully communicated information", *Acta Electronica Sinica*, Vol.38, No.3, pp.664–667, 2010. (in Chinese)

[4] X.J. Wu, Z.Z. Yang, M. Zhao, "A uniform searching particle swarm optimization algorithm", *Acta Electronica Sinica*, Vol.39, No.6, pp.1261–1266, 2011. (in Chinese)

[5] Y. Zhang, D.W. Gong, Y.Q. Ren, J.H. Zhang, "Barebones multi-objective particle swarm optimizer for constrained optimization problems", *Acta Electronica Sinica*, Vol.39, No.6, pp.1436–1440, 2011. (in Chinese)

[6] Y. Tian, D.Y. Liu, "A hybrid particle swarm optimization method for flow shop scheduling problem", *Acta Electronica Sinica*, Vol.39, No.5, pp.1087–1093, 2011. (in Chinese)

[7] J. Kennedy, "Bare bones particle swarms", *Proc. of IEEE Swarm Intelligence Symposium*, Indianapolis, Indiana, USA, pp.80–87, 2003.

[8] M.G.H. Omran, A.P. Engelbrecht, A. Salman, "Bare bones differential evolution", *European Journal of Operational Research*, Vol.196, No.1, pp.128–139, 2009.

[9] R.A. Krohling, E. Mendel, "are bones particle swarm optimization with Gaussian or Cauchy jumps", *Proc. of IEEE Congress on Evolutionary Computation*, Trondheim, Norway, pp.3285–3291, 2009.

[10] H. Zhang, D.D. Kennedy, G.P. Rangaiah, A. Bonilla-Petriciolet, "Novel bare-bones particle swarm optimization and its performance for modeling vapor-liquid equilibrium data", *Fluid Phase Equilibria*, Vol.301, No.1, pp.33–45, 2011.

[11] M. Clerc, J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", *IEEE Transactions on Evolutionary Computation*, Vol.6, No.1, pp.58–73, 2002.

[12] J.A. Nelder, R. Mead, "A simplex method for function minimization", *Computer Journal*, Vol.7, No.4, pp.308–313, 1965.

[13] C.C. Hsu, C.H. Gao, "Particle swarm optimization incorporating simplex search and center particle for global optimization", *Proc. of IEEE Conference on Soft Computing in Industrial Applications*, Muroran, JAPAN, pp.26–31, 2008.

[14] F. Wang, Y.H. Qiu, "A novel particle swarm algorithm using the simplex method operator", *Journal of Information and Control*, Vol.34, No.5, pp.517–522, 2005. (in Chinese)

[15] J.E. Dennis Jr, D.J. Woods, "Optimization on microcomputers: The Nelder-Meade simplex algorithm", in: A. Wouk, *New Computing Environments: Microcomputers in Large Scale Computing*, SIAM, Philadelphia, USA, pp.116–122, 1987.

WANG Panpan was born in 1982. He is currently pursuing the doctoral degree at the School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, China. His research interests include intelligence optimization and fault diagnosis. (Email: wpp2011@126.com)

SHI Liping was born in 1964. She received the Ph.D. degree from China University of Mining and Technology in 2001. She is a professor at School of Information and Electronic Engineering, China University of Mining and Technology. Her research interests include intelligence optimization, fault diagnosis and artificial intelligence. (Email: shiliping98@126.com)

ZHANG Yong was born in 1979. He received his Ph.D. degree from China University of Mining and Technology in 2009. He is a lecturer at School of Information and Electronic Engineering, China University of Mining and Technology. His research interests include intelligence optimization and control. (Email: yongzh401@126.com)

HAN Li was born in 1977. She received Ph.D. degree from Southeast University in 2005. She is currently an associate professor at School of Information and Electrical Engineering, China University of Mining and Technology. Her research interests include fault diagnosis and artificial intelligence. (Email: dannyli717@sohu.com)