

HPSO Algorithm with High Speed Convergent based on Particle Health Degree

Qibing Jin, Kewen Wang*, Zhenxing Zhao and Liting Cao

Institute of Automation, Beijing University of Chemical Technology, Beijing 100029, China

Received: 7 Aug. 2013, Revised: 9 Nov. 2013, Accepted: 10 Nov. 2013

Published online: 1 Jul. 2014

Abstract: Standard particle swarm optimization algorithm has two drawbacks in engineering application when particles dimension was high; first is premature convergent and second is low convergent speed. Counting these drawbacks we proposed a novel algorithm with high convergent speed in high dimensional search place based on particle health degree, and we provided particle health degree concept and computation method. The algorithm through dynamic monitoring particle health when the particle health value was lower than given threshold value, we separately use mutation operation on these particles. This method can not only protect the health particles keep searching the optimum value but also therapy the ill-health particles and enhance the ability of searching optimum value and jumping out the local optimum. We used many benchmark functions to test our algorithm, and compete with Standard PSO algorithm and nonlinear inertia weight variation (WPSO). Test results show that the algorithm we proposed has higher convergent speed and searching efficiency.

Keywords: HPSO, oscillation number, particle stagnation number, particle health degree

1 Introduction

Particle swarm optimization algorithm was a branch of evolutionary computation, it was a global random optimization algorithm based on iteration, first, it was proposed by American scholars, Kennedy and Eberhart in 1995 [1]. Compared with other evolutionary algorithms, PSO algorithm was easy to achieve and has lower space and lower time complexity. It was widely used in various optimization and control systems [2,3,4]. However basic PSO algorithm exists low searching efficiency, easy to stagnation in the local optimum and premature convergent and other issues. Counting these problems related scholars have proposed many improved algorithm. Shi et al. introduced the inertia weight (w) based on the basic PSO algorithm, this method can better retention of particles on their own speed, and this method called standard PSO algorithm. Around the improvement strategies of inertia weight, Shi et al. proposed a method to improve standard PSO algorithm based on variation inertia weight [6]. Chatterjee et al. proposed an inertia weight nonlinear reduced method [7], and Clerc proposed compression factor conception etc. [8]. Such improved algorithms can certain degree melioration the search performance and enhance the convergent speed of PSO

algorithm. But when the search space was high dimension, these methods also existed below disadvantages: low convergent speed, easy to stagnation in the local optimum.

In recent years, many new types PSO algorithms came to the fore. Meimei Zhu et al. through chaotic initialize and detected stagnation times of particles, and use global mutation strategy to the whole swarm, this method improved the convergent speed, obtained good results [9]. Lianguo Wang et al. purposes a hybrid algorithm of Particle Swarm Optimization (PSO) and Artificial Fish Swarm Algorithm (AFSA) by combining the advantages of PSO algorithm and AFSA algorithm [10]. Hybrid algorithm divided the swarm into two sub-groups. In each iteration one sub-group evolved using PSO algorithm, the other sub-group evolved using AFSA, and two algorithms shared the information of groups extremum. Ximing Liang et al. analyzed the contradiction of the global exploration and convergent speed of particle swarm optimization with dimension mutation operator, and all improved algorithm (WPSO) was proposed by modifying PSO with dimension mutation based on dynamical inertial weight vector [11]. In the proposed algorithm, the concept of dimension diversity was defined

* Corresponding author e-mail: heaven996@hotmail.com

and inertia weight vector will be updated dynamically according to dimension diversity. The mutation operated on dimension whose dimension diversity is the worst. Yanmin Liu et al. through analyzing the relationship between swarm diversity and local optimum presented an improved particle swarm optimizer based on dynamic neighbor topology (DPSO for short) [12]. In DPSO the neighbor of each particle was dynamically constructed at several iterations which increased the swarm diversity and improved the ability to escape from local optimum. Min Zhou proposed a novel method called (k, l) PSO [13]. This method chose one of the top k particles as the global best particle according to the roulette strategy and tuned the inertia weight value according to the distance between the current particle and the global best particle. Yuehong Liu et al. proposed an improved PSO based on the diversity of particle symmetrical distribution (sdPSO) is developed [14]. Through the research of the spatial distribution of particles it can be found that the convergent probability to the global optimum solution was greatly improved with more symmetrical particle distribution surrounding the optimum solution of particles. A diversity population function was proposed and an adjustment algorithm for the diversity was introduced into the basic PSO. The spatial distribution of particles varies between asymmetry and symmetry repeatedly while the population diversity was adjusted continually which made the improved algorithm search in a wider range. Shutao Li et al. proposed a hybrid global optimization strategy combining PSOs with a modified.

Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [13]. The modified BFGS method was integrated into the context of the PSOs to improve the particles' local search ability. In addition, in conjunction with the territory technique, a reposition technique to maintain the diversity of particles was proposed to the hybrid strategy was that it could effectively find multiple local solutions or global solutions to the multimodal functions in a box-constrained space. Based on these local solutions, a reconstruction technique can be adopted to further estimate better solutions. Jiuzhong Zhang et al. proposed a Multi-Swarm Self-Adaptive and Cooperative Particle Swarm Optimization (MSCPSO) based on four sub-swarms [16]. In this method several strategies were employed to avoid falling into local optimum, improved the diversity and achieved better solution. Particles in each sub-swarm shared the only global historical best optimum to enhance the cooperative capability. Besides, the inertia weight of a particle in each sub-swarm was modified, which was subject to the fitness information of all particles, and the adaptive strategy was employed to control the influence of the historical information to create more potential search ability. To effectively keep the balance between the global exploration and the local exploitation, the particle in each taken advantage of the shared information to maintain cooperation with each other and guided its own evaluation. On the other hand, in order to increase the diversity of the particles and avoid

falling into a local optimum, a various diversity operation was adopted to guide the particles to jump out of the local optimum and achieve the global best position smoothly. Hamidreza Modares et al. proposed a hybrid algorithm by integrating an improved particle swarm optimization (IPSO) with successive quadratic programming (SQP), namely IPSO-SQP for solving nonlinear optimal control problems [17]. The particle swarm optimization (PSO) was showed to converge rapidly to a near optimum solution, but the search process will become very slow around global optimum. On the contrary, the ability of SQP was weak to escape local optimum but could achieve faster convergent speed around global optimum and the convergent accuracy can be higher. Hence, in the proposed method, at the beginning stage of search process, a PSO algorithm was employed to find a near optimum solution. In this case, an improved PSO (IPSO) algorithm was used to enhance global search ability and convergent speed of algorithm. When the change in fitness value was smaller than a predefined value, the searching process was switched to SQP to accelerate the search process and found an accurate solution. In this way, this hybrid algorithm may find an optimum solution more accurately.

In this paper, we introduced the particle health degree conception, and given particle health degree definition and computation method. We dynamically computed particle health degree simultaneously and taken mutation operation with low health degree particles. Particle health degree was a comprehensive reflection of particles in the iterative process in many ways. When the health of the single particle reduced to a certain threshold value, particles mutation began. This was more targeted while particles mutation. It can not only avoid normal particles iterative evolution process be force interrupted, but also through mutation algorithm to effectively reduce the occurrence of particle precocious stagnation, enhanced the diversity of particles and improved the global search ability of PSO. Many simulation experiments showed that the method in our paper had faster convergent speed and higher search efficient when dimensions of particle were high.

2 Standard PSO Algorithm

The mathematical description of Standard Particle Swarm Optimization (SPSO) is: in the search place which dimension was D , each particle was looked as a node in place. Assumed the swarm was set by m particles, $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ was the i -th particle D -dimension position vector, where $i = 1, 2, \dots, m$, $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ was the i -th particle velocity vector. $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$ was the best position vector where the i -th particle had searched, $P_g = (P_{g1}, P_{g2}, \dots, P_{gD})$ was the most optimal position where the whole swarm had searched.

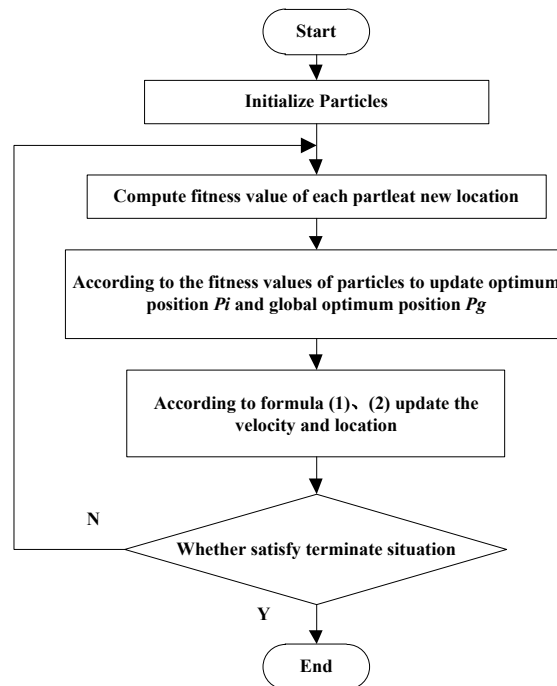


Fig. 1: Standard PSO Algorithm Flow Chart

Put x_i into the objective function to calculate the fitness value. For the $k + 1$ -th iteration, each particle according to the formulas below updated its speed and position.

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot \text{rand}_1 \cdot (P_i - x_i^k) + c_2 \cdot \text{rand}_2 \cdot (P_g - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

Where $i = 1, 2, \dots, m$, w was inertia weight, c_1 and c_2 were learning factors, rand_1 and rand_2 were random numbers, when $w = 1$, was basic PSO algorithm, Figure 1 showed standard PSO algorithm flow chat.

Because of the standard PSO algorithm did not have any actual mechanism to control particle velocity. In order to prevent flight speed too fast that generated strongly shocked and hard to converge we introduced $V_{\max} = (V_{\max 1}, V_{\max 2}, \dots, V_{\max D})$ as the max speed velocity. In each iteration, particle according to formulas below updated their new position and speed.

$$v_{id}^{k+1} = \begin{cases} V_{\max d} & \bar{v}_{id}^{k+1} > V_{\max d} \\ \bar{v}_{id}^{k+1} & -\bar{v}_{id}^{k+1} \leq \bar{v}_{id}^{k+1} \leq V_{\max d} \\ -V_{\max d} & \bar{v}_{id}^{k+1} < -V_{\max d} \end{cases} \quad (3)$$

Where $i = 1, 2, \dots, m$, $d = 1, 2, \dots, D$.

3 Inertia Weight Variation PSO Algorithm

At the beginning of the inertia weight variation PSO algorithm, the larger w was helpful to enhance the global search ability of algorithm, and jumped out the local optimum. At the later of the optimization process, the smaller w was helpful to improve the local search ability, and made the algorithm converge. Decreasing w by the parabolic rule can more effectively control algorithm exploration ability and exploitation ability. Through the w fallen by parabolic rule, it can be easily control the algorithm ability of exploration and exploitation. Base on above, the inertial weight variation strategy was proposed. Looked inertia factor w in the standard PSO speed updated formula (4) as iteration time t function. Changing as:

$$w = w_{\max} - (w_{\max} - w_{\min}) \left(\frac{t}{t_{\max}} \right)^2 \quad (4)$$

Where t_{\max} was the max iteration, w_{\max} , w_{\min} were the max and min inertia weight.

The Status of the Particle Oscillation Judgment Method.

Through the observation to particle trajectory we found when particle evolution stagnation happened, often appeared the particle trajectory at one dimension oscillation condition (Shown as Fig. 2).

Trajectory of Particles in movement process can be usually abstracted six statuses as shown in figure 3. It can

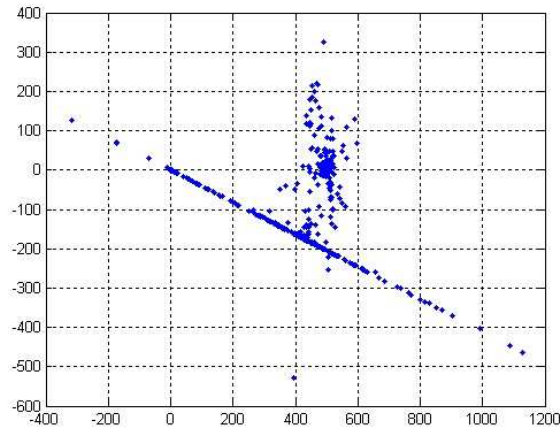


Fig. 2: Particle Trajectories in PSO Optimization Process

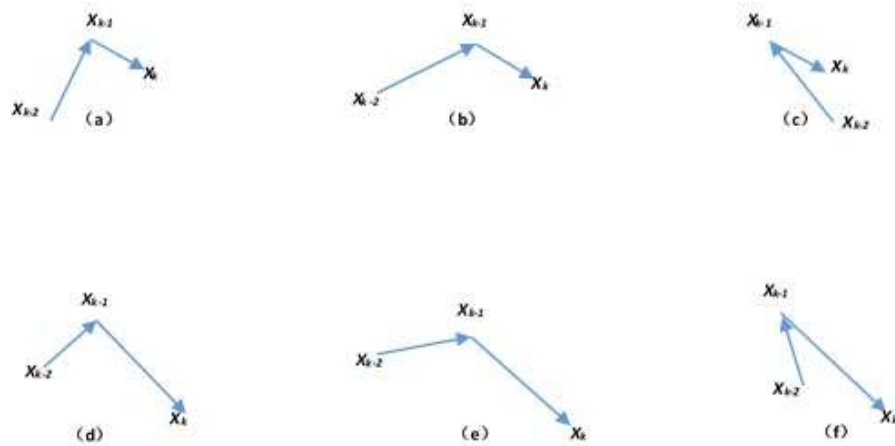


Fig. 3: Particle Trajectory Schemes

be concluded that judge particle trajectory appear oscillation status method:

When particles moving, if continuous two generations were all satisfied formula 5, we can judge particles trajectory appear oscillation at this dimension. Meaning that if particle trajectory often showed like Fig. 3(c) or Fig. 3(f), we can judge particle appear oscillation at this dimension.

$$\frac{\max(|x_{k-2} - x_{k-1}|, |x_k - x_{k-1}|)}{|x_k - x_{k-2}|} > 3 \tag{5}$$

where x_i was the location in situation i .

When particle trajectory appeared oscillation, the optimization process often happened stagnation, or premature convergent happened. We can use the method below to identify particle happened opening oscillation and convergent oscillation.

If $x_k - x_{k-1}$ and $x_k - x_{k-2}$ were different signals, we can judge particles happened convergent oscillation.

If $x_k - x_{k-1}$ and $x_k - x_{k-2}$ were same signals, we can judge particles happened divergent oscillation.

The number of particles oscillation and particles stagnation.

4 Deterministic particle oscillation number N_{osc}

When particle first appeared oscillation, it was said that particle in moving process, their movement positions were twice satisfied by the formula 5, we considered the oscillation number N_{osc} as 1, or considered as 0. If in situation $N_{osc} > 0$, in the next several continuous generations, once formula 5 was satisfied, the N_{osc}

number was added by 1, until the formula 5 was not satisfied or forced to mutation.

5 Deterministic particle stagnation number

N_s

In each particle iteration process, we used a counter to record the stagnation number of particle currently. When detected the optimum didn't change to the last optimum, the stagnation number was added by 1, or clear to 0.

Particle health degree can be regard as health merit of single particle searching optimum process. The main principle was, if particles in every iteration is closer and closer to the optimum, their health degrees will be increased, otherwise if particle appeared stagnation or oscillation their health degrees will be decreased. After mutation particles health degree will be reset.

Particle health degree defined as the function combined with particle stagnation number N_s and particle oscillation number N_{osc} . We calculated the function like the method below:

$$H_{particle} = 100 - \min(w_s N_s + w_{osc} N_{osc}, 100) \quad (6)$$

Where w_s and w_{osc} were the weight of stagnation number and oscillation number respectively. $w_s = 3$ and $w_{osc} = 0.8$ HPSO that we proposed in this paper.

6 Fast convergent PSO optimization algorithm based on health

By analyzing the situation that particles in high level dimension, particles were easily fallen in local optimum. We monitored particles health degree real-time and counted low particles health degrees made mutation to solve this problem. Firstly, we need to according to particular matter to ascertain a suitable threshold value H_{th} , when detected certain particle health degree lower than the threshold value H_{th} , through re-assignment the local optimum (P_i in formula 1) to change the particle original movement trajectory. Shown as formula 7. Thus changed the velocity vector that particles updated, promoted the particles to escape from local optimal.

$$P_i = \text{rand} \times (P_g - x_i) \quad (7)$$

Where P_i was the i -th particle local optimum location, P_g was particle swarm global optimum location, X_i was the i -th particle location. The flowchart of standard PSO algorithm was shown in Fig. 4.

The pseudo-code of fast convergent PSO based on health degree was proposed in this paper described below:
 Step 1. Random Initialize velocities and positions which particles in swarm.

Step 2. While (not satisfied terminate condition).

Step 3. For Each Particle.

Step 4. According to criterion function calculate fitness value of each particle.

Step 5. According to formula(1),(2),(3)update velocities and positions of particles.

Step 6. According to formula (5) calculate oscillation numbers of particles.

Step 7. Calculating stagnation particles numbers.

Step 8. According to formula (6) update health degrees of each particles.

Step 9. If (particle health degree \leq threshold value H_{th}) then

Step 10. Making the particles mutation, changing its local optimization position.

Step 11. Resetting the particles health degree was 100, which had been mutation.

Step 12. End If

Step 13. End for

Step 14. End While

Step 15. End

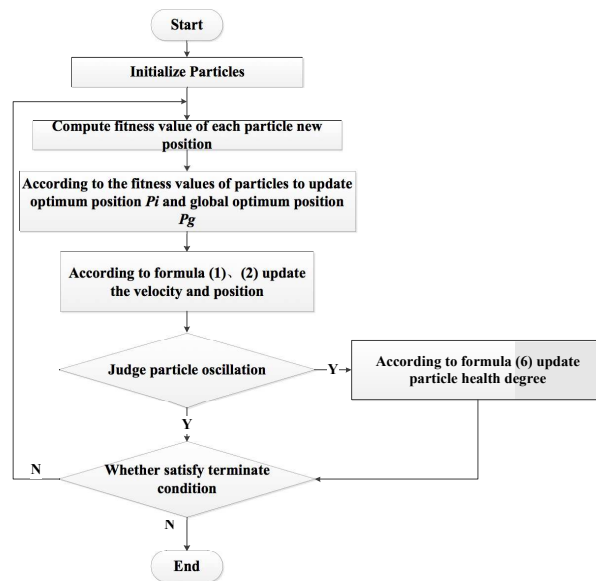


Fig. 4: HPSO Algorithm Flow Chart

7 Simulation Experiments

In order to verify the validity of the HPSO algorithm, we conducted a large number of numerical experiments. To test the effect of the HPSO algorithm in higher dimension status, we chose Ackley, Cigar, Ellipse, Griewank, Rastrigrin, Noncontinuous Rastrigrin, Rosenbrock, Quadric, Sphere these ten variable dimension benchmark

Table 1: Benchmark Functions

Function Name	Functional Expression	Region of $i = 1, 2, \dots, n$	Extremum Local	Optimum Location	Optimum
Ackley	$20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{-\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)}$	$[-32, 32]$	multi	$(0, \dots, 0)$	0
Cigar	$x_1^2 + 10^4 \sum_{i=2}^n x_i^2$	$[-5, 5]$	none	$(0, \dots, 0)$	0
Ellipse	$\sum_{i=1}^n \left[\left(10^4 \frac{i-1}{n-1} \right) x_i^2 \right]$	$[-5, 5]$	none	$(0, \dots, 0)$	0
Griewank	$\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]$	multi	$(0, \dots, 0)$	0
NC-R	$\sum_{i=1}^n (y_i^2 - 10 \cos(2\pi y_i) + 10)$ $y_i \begin{cases} x_i & x_i < 0.5 \\ \frac{\text{round}(2x_i)}{2} & x_i \geq 0.5 \end{cases}$ $i = 1, 2, \dots, n$	$[-5.12, 5.12]$	multi	$(0, \dots, 0)$	0
Quadric	$\sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	none	$(0, \dots, 0)$	0
Rastrigrin	$10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	$[-5.12, 5.12]$	multi	$(0, \dots, 0)$	0
Rosenbrock	$\sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-2.5, 2.5]$	multi	$(1, \dots, 1)$	0
Sphere	$\sum x_i^2$	$[-100, 100]$	none	$(0, \dots, 0)$	0

Table 2: Parameters used in the test

Learning factor (c_1)	1.5	Particle number(N)	20
Learning factor (c_2)	1.5	inertia weight (w)	0.5
precision (eps)	0.0001	Max iteration time (M)	3000

functions (Table 1) and repeated test, competed them with SPSO, WPSO these two common particle swarm optimization algorithms.

To make a fairer comparison among the algorithms, pre-generate a set of initial values and use it to initialize the particles before we run algorithms each time, and thus effectively eliminating the efficiency of the algorithm on different particle initial position caused the differences. Particle numbers N of SPSO, WPSO and HPSO were 30, inertia weight w was 0.5, learning factors c_1 and c_2 were both 1.5. The inertia weight w of WPSO algorithm was decrease 0.9 to 0.4. In HPSO algorithm we set the health degree threshold 85, particle oscillation number weight w_{osc} was 0.8, particle stagnation number w_s was 3. The public parameters of the three algorithms were shown in table 2.

7.1 Experiment Design

Firstly, we fixed particle dimension when achieved certain precision (take precision $eps = 0.0001$), the iteration times that SPSO, WPSO and HPSO needed. To avoid the searching optimum process not fall into drop-dead halt, we set when particles iteration times was more than 1000, algorithms were ended, and returned 0. It was mean that

in the provision of iteration times cannot reach required accuracy. In order to reduce the influence of random search, each algorithm was run several times and calculated statistics. The results of the comparisons were shown in table 2. Where particle numbers N of SPSO, WPSO and HPSO were 30, inertia weight w was 0.5, learning factors C_1 and C_2 were 1.5. The inertia weight w of WPSO algorithm was decrease 0.9 to 0.5. Mean values and standard deviations in Tables were only counted test result the particles search accuracy that reach the accuracy of the provisions, not included algorithm achieves maximum iterating times but didn't reach the required accuracy of test results.

7.2 The convergence test

Convergence test was the test of convergence of the three algorithms in the specified number of iterations. In this paper we fixed the dimensions of the particles was 10, the test results as shown in figure 5(a) to figure 5(i).

From the figures we can see, except the Rosenbrock function, for the other benchmark functions, the HPSO algorithm can search the global optimum in less iterations.

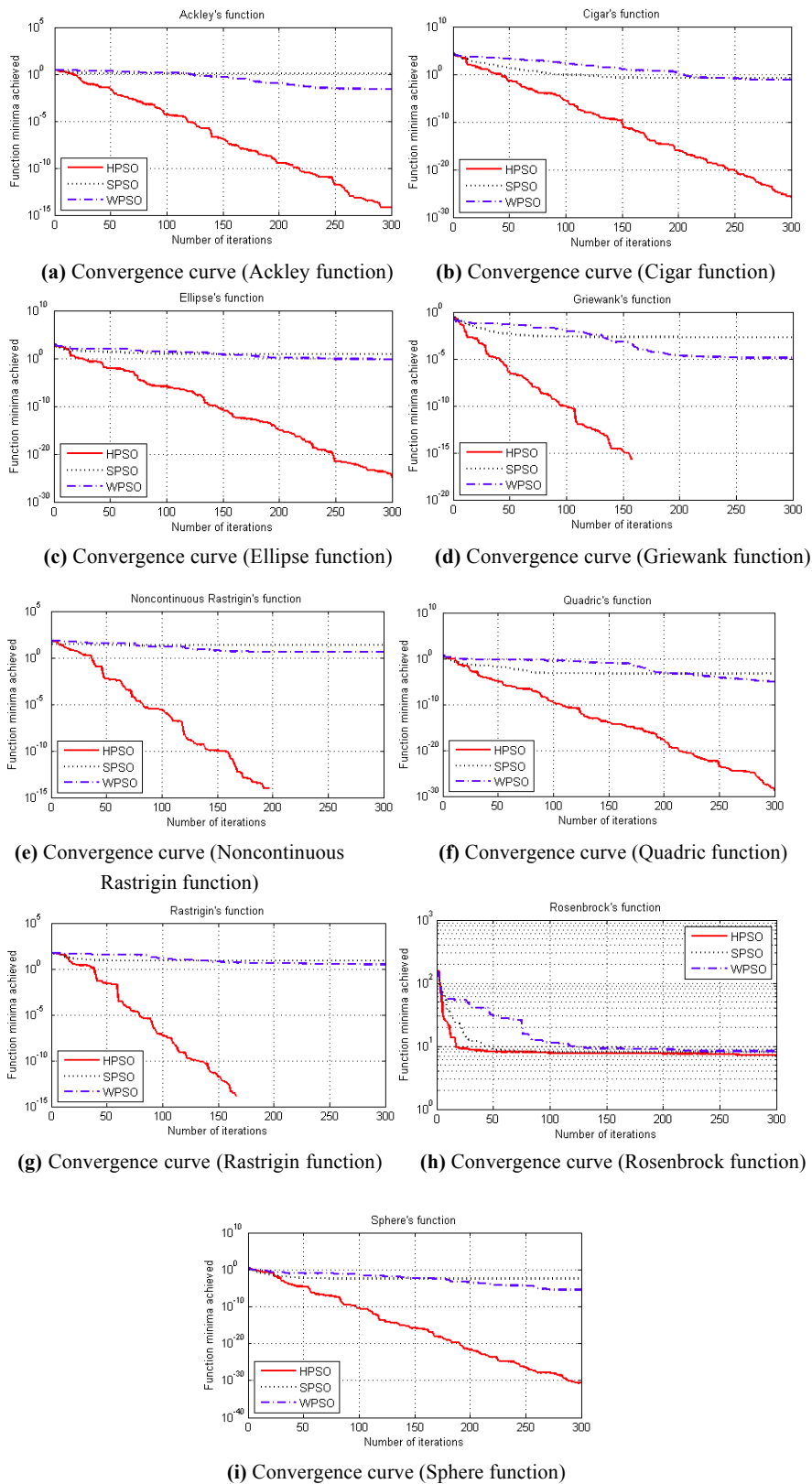


Fig. 5: Convergence curve

Table 3: Results for 5-Dimension problems

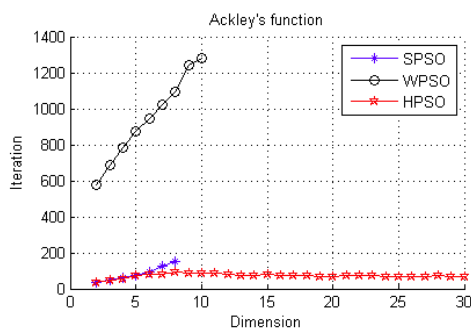
Benchmark	PSOs	<i>Best</i>	<i>Worst</i>	<i>Mean</i>	<i>Std.</i>	<i>SR</i>
Ackley	SPSO	-8.88E-16	1.65E+00	8.23E-02	3.68E-01	95%
	WPSO	-8.88E-16	6.22E-15	2.84E-15	1.40E-15	100%
	HPSO	-8.88E-16	-8.88E-16	-8.88E-16	0.00E+00	100%
Cigar	SPSO	2.50E-308	1.12E-285	5.58E-287	0.00E+00	100%
	WPSO	2.07E-176	2.50E+01	1.25E+00	5.59E+00	95%
	HPSO	7.67E-302	5.71E-289	6.30E-290	0.00E+00	100%
Ellipse	SPSO	6.24E-314	2.52E-290	1.26E-291	0.00E+00	100%
	WPSO	2.09E-173	3.11E-164	1.65E-165	0.00E+00	100%
	HPSO	1.28E-302	7.05E-288	3.93E-289	0.00E+00	100%
Griewank	SPSO	0.00E+00	2.22E-16	1.11E-17	4.97E-17	100%
	WPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
NC-R	SPSO	1.00E+00	1.00E+01	3.20E+00	2.09E+00	0%
	WPSO	0.00E+00	3.00E+00	1.40E+00	7.54E-01	5%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
Quadric	SPSO	8.82E-316	5.82E-300	2.96E-301	0.00E+00	100%
	WPSO	1.96E-174	4.29E-165	2.29E-166	0.00E+00	100%
	HPSO	3.48E-307	1.75E-287	8.80E-289	0.00E+00	100%
Rastrigrin	SPSO	0.00E+00	6.96E+00	3.33E+00	1.78E+00	5%
	WPSO	0.00E+00	3.98E+00	1.24E+00	1.11E+00	30%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
Rosenbrock	SPSO	0.00E+00	3.93E+00	1.97E-01	8.79E-01	95%
	WPSO	0.00E+00	4.19E-30	4.19E-31	1.29E-30	100%
	HPSO	1.05E-30	5.71E-10	4.74E-11	1.49E-10	100%
Sphere	SPSO	1.89E-311	7.94E-299	4.00E-300	0.00E+00	100%
	WPSO	4.42E-177	4.58E-164	2.30E-165	0.00E+00	100%
	HPSO	4.52E-306	5.34E-286	2.67E-287	0.00E+00	100%

Table 4: Results for 10-Dimension problems

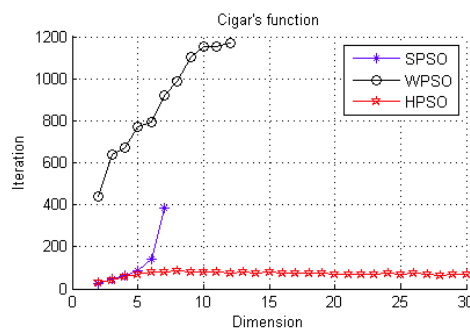
Benchmark	PSOs	<i>Best</i>	<i>Worst</i>	<i>Mean</i>	<i>Std.</i>	<i>SR</i>
Ackley	SPSO	2.44E-03	2.02E+00	5.97E-01	7.53E-01	0%
	WPSO	2.81E-06	1.61E-03	2.78E-04	4.25E-04	55%
	HPSO	-8.88E-16	-8.88E-16	-8.88E-16	0.00E+00	100%
Cigar	SPSO	1.15E-01	2.57E+01	4.15E+00	6.14E+00	0%
	WPSO	1.57E-08	3.66E-01	1.98E-02	8.16E-02	45%
	HPSO	6.68E-291	3.45E-272	1.95E-273	0.00E+00	100%
Ellipse	SPSO	8.71E-02	6.79E+00	2.70E+00	2.26E+00	0%
	WPSO	3.43E-09	6.96E+01	3.48E+00	1.56E+01	45%
	HPSO	8.35E-288	2.61E-273	1.30E-274	0.00E+00	100%
Griewank	SPSO	1.81E-05	6.69E-03	1.02E-03	1.50E-03	15%
	WPSO	3.96E-11	7.40E-03	3.70E-04	1.65E-03	95%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
NC-R	SPSO	4.00E+00	2.10E+01	1.16E+01	4.75E+00	0%
	WPSO	5.00E+00	1.00E+01	7.35E+00	1.66E+00	0%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
Quadric	SPSO	6.52E-04	8.54E-02	2.37E-02	2.30E-02	0%
	WPSO	1.04E-07	1.46E-03	1.72E-04	3.86E-04	75%
	HPSO	5.85E-292	1.23E-277	7.33E-279	0.00E+00	100%
Rastrigrin	SPSO	2.12E+00	1.85E+01	1.02E+01	4.18E+00	0%
	WPSO	9.95E-01	1.29E+01	5.97E+00	3.16E+00	0%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
Rosenbrock	SPSO	5.43E+00	9.92E+00	8.06E+00	1.45E+00	0%
	WPSO	3.14E-02	7.52E+00	1.97E+00	1.92E+00	0%
	HPSO	1.34E-10	6.91E+00	4.91E+00	1.60E+00	5%
Sphere	SPSO	3.53E-04	2.11E-02	5.73E-03	6.53E-03	0%
	WPSO	3.17E-12	1.96E-06	2.23E-07	5.11E-07	100%
	HPSO	1.67E-293	5.89E-278	2.96E-279	0.00E+00	100%

Table 5: Results for 30-Dimension problems

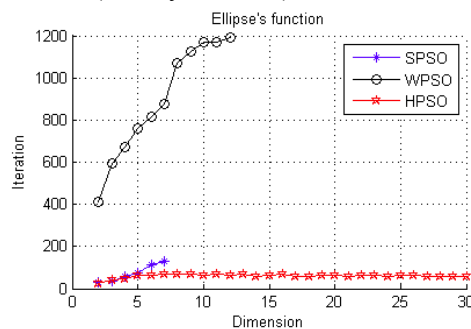
Benchmark	PSOs	Best	Worst	Mean	Std.	SR
Ackley	SPSO	1.08E+00	2.11E+00	1.56E+00	2.68E-01	0%
	WPSO	7.36E-02	8.34E-01	3.60E-01	2.02E-01	0%
	HPSO	-8.88E-16	-8.88E-16	-8.88E-16	0.00E+00	100%
Cigar	SPSO	1.53E+03	1.02E+04	5.51E+03	2.63E+03	0%
	WPSO	2.85E+01	2.91E+02	7.99E+01	6.26E+01	0%
	HPSO	0.00E+00	3.60E-321	3.36E-322	0.00E+00	100%
Ellipse	SPSO	3.17E+01	2.46E+02	1.54E+02	6.88E+01	0%
	WPSO	1.52E+00	9.38E+01	1.62E+01	2.62E+01	0%
	HPSO	0.00E+00	3.11E-322	1.48E-323	0.00E+00	100%
Griewank	SPSO	1.35E-02	8.08E-02	3.55E-02	1.63E-02	0%
	WPSO	4.37E-03	3.83E-02	1.87E-02	8.48E-03	0%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
NC-R	SPSO	2.77E+01	1.08E+02	6.42E+01	2.07E+01	0%
	WPSO	8.14E+00	7.11E+01	3.57E+01	1.58E+01	0%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
Quadric	SPSO	5.71E-01	2.37E+00	1.44E+00	5.16E-01	0%
	WPSO	5.06E-02	2.55E+00	7.31E-01	5.13E-01	0%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
Rastrigrin	SPSO	5.27E+01	1.41E+02	9.52E+01	2.18E+01	0%
	WPSO	1.07E+01	5.16E+01	2.88E+01	1.03E+01	0%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%
Rosenbrock	SPSO	4.34E+01	1.49E+02	7.84E+01	2.70E+01	0%
	WPSO	2.84E+01	3.00E+01	2.90E+01	3.99E-01	0%
	HPSO	1.06E-09	2.88E+01	2.13E+01	1.22E+01	20%
Sphere	SPSO	2.59E-01	1.47E+00	6.89E-01	3.24E-01	0%
	WPSO	1.85E-03	5.29E-01	1.46E-01	1.32E-01	0%
	HPSO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	100%



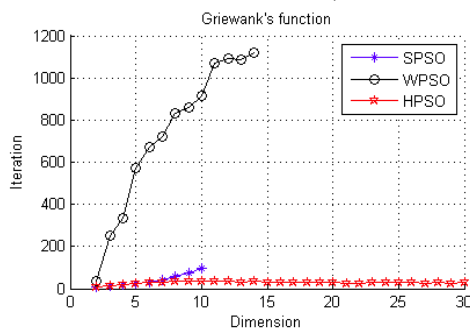
(a) Convergence speed contrast (Ackley function)



(b) Convergence speed contrast (Cigar function)



(c) Convergence speed contrast (Ellipse function)



(d) Convergence speed contrast (Griewank function)

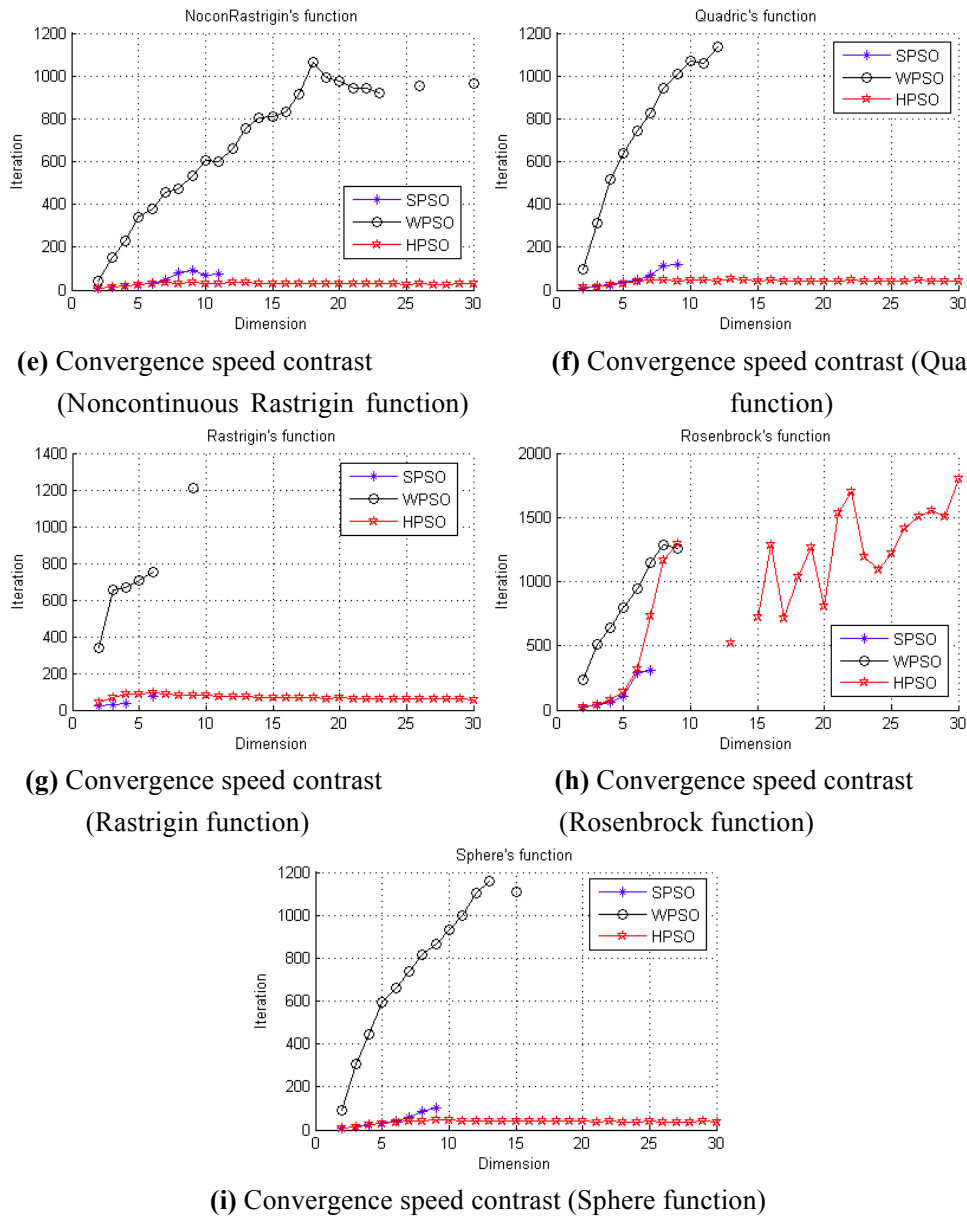


Fig. 6: Convergence speed contrast

7.3 Stable Test

Because of the randomness in the PSO algorithm, we can't judge which algorithm was better in one test. We need to test the stability of the three algorithms. In this test, for each algorithm, we run each benchmark function 20 times, and then obtained the experiment results by statistics. Table 3 to Table ?? was experiment results corresponding to the dimensions of the particles 5, 10, 30 respectively. In these tables we listed the statistics were: best value (*Best*), worst value (*Worst*), mean value

(*Mean*), variance (*Std*) and success rate (*SR*). The success rate was shown in formula 8.

$$SR = \frac{N_{\text{succed}}}{N_{\text{total}}} \times 100\% \tag{8}$$

Where N_{total} was the total test times. N_{succed} was the success times. In the specified iterations (M), the optimal accuracy was higher than the preset accuracy (ϵ) was considered successful, otherwise judged to be failures.

From tables above we can see, for all the benchmark test functions, the success rate of HPSO algorithm was higher than the other two algorithms, and in most of tests,

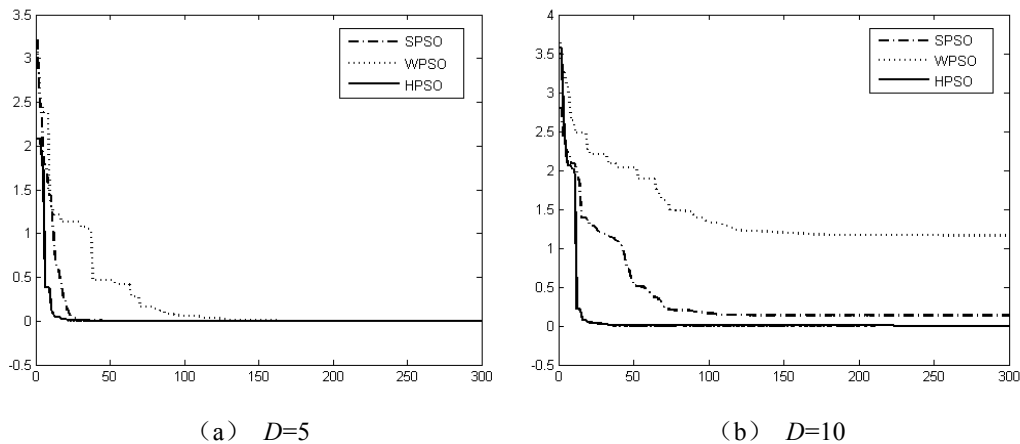


Fig. 7: Comparison optimization Efficiency in Different Dimension

the HPSO algorithm success rate were 100%. Some standard deviation in the tables were zero was due to the computer round-off errors.

7.4 Convergence speed test

Increased problem dimension from 2 to 30, and tested these three algorithms in each dimension. The searching precision ϵ was 0.0001. In each dimension we tested 10 times. The results were shown in Fig. 6(a) to Fig. 6(i).

In the figures above we can see, HPSO proposed by this paper was better than the other two algorithms. Along with the dimension increasing SPSO and WPSO cannot convergent in limited iterations. However HPSO can convergent in limited iterations and precision achieve 0.0001 except benchmark functions Rosenbrock and Schwefel. We use high-dimension and multi-modal function Ackley as an instance, comparison these three algorithm optimization efficiency in different dimensions. Shown as Fig. 7.

Fig. 7 (a, b) were the alteration lines that particles global optimum in optimization searching; three optimization algorithms tested to Ackley function in dimension 5 and 10 respectively. Table 4 shown in different dimensions condition these three algorithms needed minimum iterations numbers that achieve precision 0.0001(each algorithm tested 20 times mean values).

In order to let the tests results more universality and more Intuitive, we tested HPSO algorithm in higher dimension. Shown as Fig. 8, we used Ackley function to tested HPSO algorithm. Where X-axis was particle dimension, and sampled twice added. Y-axis was the iteration that achieved the precision 0.0001 needed. Because of randomness of the algorithm, we test every dimension simple point 10 times and calculated mean value.

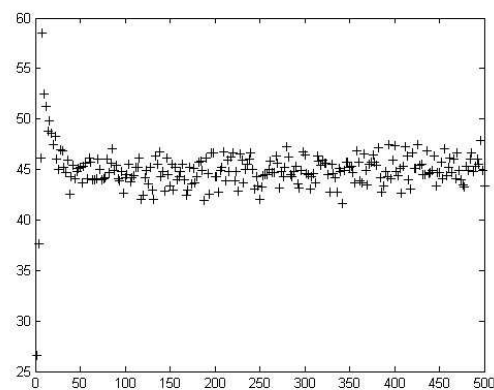


Fig. 8: Iterations of Achieving Fixed Precision Algorithm Needed

We can see from the comparison results above, along with dimension increasing, iterations increasing slowly that the HPSO algorithm convergent to fixed precision 0.0001, almost invariant. Yet in the dimension was 8 and dimension was 10 respectively, the standard particle swarm optimization and weight inertia variation optimization cannot convergent in 3000 iteration times. We can see that in high dimension condition, SPSO and WPSO were easily dropped into local optimum, as a result they were all untimely convergence. HPSO proposed by us can still jump out local optimum and search global optimum in limited iterations times.

8 Conclusions and Forecast

There were three improvements in this paper: First, proposed health degree conception. Second, proposed a

method of computing health degree. Third, in optimization searching, we monitored particle health degree singly, and timely singly mutated account to the particle which health degree lower than the given threshold value. Given a random turbulence to the particle at all dimensions, and changed particles movement trajectory. Jumped out from local optimal solution, effectively reduced the invalid iteration, improved the convergent speed and searching accuracy of particle swarm optimization. At the same time also can ensure optimization process of the high health degree particles didn't interrupt, and further improved the searching efficiency of the whole group.

The experiments proved that in most high dimension test functions, HPSO algorithm proposed by this paper was significantly higher than the other two algorithms in the ways of convergent accuracy and convergent speed.

Because of the choice of health degree threshold value in a certain extent influence the performance of the algorithm, how to choice threshold value will be the future research subject. Meanwhile HPSO algorithm proposed by us could fusion with other algorithms and further improved the performance of fusion algorithm.

Acknowledgement

This work is partially supported by the financial support of the National High-Tech Research and Development Plan of China (Grant 2008AA042131) and the National Grand Fundamental Research 973 Program of China (Grant 2007CB714300) and partly by the Automation Institute Beijing University of Chemical Technology. The authors are grateful to the anonymous reviewers for their valuable recommendations.

References

- [1] J. Kennedy, R. Eberhart, Particle swarm optimization, in Neural Networks, 1995. Proceedings., IEEE International Conference on, **4**, 1942-1948 (1995).
- [2] H. Shayeghi, H. A. Shayanfar, S. Jalilzadeh, A. Safari, Design of output feedback UPFC controller for damping of electromechanical oscillations using PSO, Energy Conversion and Management, **50**, 2554-2561 (2009).
- [3] Taher Seyed Abbas, Karimian Ali, Hasani Mohammad, A new method for optimal location and sizing of capacitors in distorted distribution networks using PSO algorithm, Simulation Modelling Practice and Theory, **19**, 662-672 (2011).
- [4] H. Xiaohui, R. C. Eberhart, S. Yuhui, Engineering optimization with particle swarm, in Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE, 53-57 (2003).
- [5] S. Yuhui, E. Russell, A modified particle swarm optimizer, IEEE International Conference on Evolutionary Computation, 69-73 (1998).
- [6] S. Yuhui, E. C. Russell, Parameter Selection in Particle Swarm Optimization, Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming, 591-600 (1998).
- [7] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Computers & Operations Research, **33**, 859-871 (2006).
- [8] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, **3**, 1957 (1999).
- [9] Z. Meimei, W. Yongping, A PSO algorithm with high speed convergence, Control And Decision, **25**, 20-24 (2010).
- [10] W. Lianguo, S. Qihong, H. Yi, Hybrid Optimization Algorithm of PSO and AFSA, Computer Engineering, **36**, 176-178 (2010).
- [11] L. Ximing, D. Shuhua, L. Wen, PSO algorithm with dynamic inertia weight vector and dimension mutation, Journal Of Computer Applications, **47**, 29-31 (2011).
- [12] L. Yanmin, Z. Qingzhen, N. Ben, Particle Swarm Optimization Algorithm Based on Dynamic Neighbor Topology Framework, Computer Engineering, **37**, 210-212 (2011).
- [13] Z. Min, Distance-based adaptive k particle swarm optimization. Computer Engineering and Applications, **47**, 43-45 (2011).
- [14] S. Yuehong, W. Jianxiang, X. Deshen, An Improved PSO Based on Diversity of Particle Symmetrical Distribution, Pattern Recognition And Artificial Intelligence, **23**, 137-143 (2010).
- [15] L. Shutao, T. Mingkui, W. T. Ivor, James Tin-Yau Kwok A Hybrid PSO-BFGS Strategy for Global Optimization of Multimodal Functions, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, **41**, 1003-1014 (2011).
- [16] Z. Jiuzhong, D. Xueming, A Multi-Swarm Self-Adaptive and Cooperative Particle Swarm Optimization, Engineering Applications of Artificial Intelligence, **24**, 958-967 (2011).
- [17] M. Hamidreza, N. S. Mohammad-Bagher, Solving nonlinear optimal control problems using a hybrid IPSO-SQP algorithm, Engineering Applications of Artificial Intelligence, **24**, 476-484 (2011).



Qibing Jin received the MS degree from Northeast Petroleum University in 1996, and the PhD degree from the department of Optics Engineering, Northeastern University in 1999. He is currently a professor in Beijing University of Chemical Technology. His

research interests are in the areas of advanced control and application in industry, process modeling and optimization, and multivariate prediction and internal model control.



Kewen Wang received the MS degree from Liaoning Normal University in 2010, and the Phd degree from Institute of Automation in Beijing University of Chemical Technology in 2013. His research interests

are in the areas of real-time database application, process modeling and optimization and system identification.



Zhenxing Zhao received the MS degree from Institute of Automation in Beijing University of Chemical Technology in 2011. His research interests are in the areas of system identification theory and its application.



Liting Cao Lecturer at Beijing Union University and Ph. D. candidate at the College of Information Science and Technology, Beijing University of Chemical Technology. She received her master degree from University of Science and Technology of China in 2005. Her research interest

covers advanced control and its industrial application.