Neurocomputing 137 (2014) 234–240

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A diversity-guided hybrid particle swarm optimization based on gradient search

Fei Han *, Qing Liu

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China

ARTICLE INFO

Article history:
Received 2 January 2013
Received in revised form
14 March 2013
Accepted 30 March 2013
Available online 18 February 2014

Keywords:
Particle swarm optimization
Diversity
Gradient search

ABSTRACT

As an evolutionary computing technique, particle swarm optimization (PSO) has good global search ability, but it is easy to make the swarm lose its diversity and lead to premature convergence. In this paper, a diversity-guided hybrid PSO based on gradient search is proposed to improve the search ability of the swarm. The adaptive PSO is first used to search the solution till the swarm loses its diversity. Then, the search process turns to a new PSO (DGPSOGS), and the particles update their velocities with their gradient directions as well as repel each other to improve the swarm diversity. Depending on the diversity value of the swarm, the proposed hybrid method switches alternately between two PSOs. The hybrid algorithm adaptively searches local minima with random search realized by adaptive PSO and performs global search with semi-deterministic search realized by DGPSOGS, and so its search ability is improved. The experimental results show that the proposed hybrid algorithm has better convergence performance with better diversity compared to some classical PSOs.

1. Introduction

In the past decades, feedforward neural networks (FNN) have been widely used in the field of machine learning [1–3]. Gradient descent algorithms such as backpropagation (BP) are one of the most popular learning algorithms for FNN. However, these gradient descent algorithms are easy to converge to local minima as well as converge very slowly [4–7]. To decrease the likelihood of being trapped into the local minima of the error surface, some methods incorporating prior information of the involved problems were proposed [4–7]. Although these algorithms improved their generalization performance greatly, they also increased the computational complexity as well as running time. Although their performance is improved, they are local search algorithm in essence.

As a global search algorithm, particle swarm optimization (PSO) is a population-based stochastic optimization technique developed by Kennedy and Eberhart [8,9]. Since PSO is easy to implement without requiring complex evolutionary operations compared to genetic algorithm (GA) [10,11], it has been widely used in many fields such as power system optimization, process control, dynamic optimization, adaptive control and electromagnetic optimization [12,13]. Although PSO has shown good performance in solving many optimization problems, it suffers from the problem of premature convergence like most of the stochastic search techniques, particularly in multimodal optimization problems [13].

To improve the performance of PSO, many improved PSOs were proposed. Passive congregation PSO (PSOPC) introduced passive congregation for preserving swarm integrity to transfer information among individuals of the swarm [14]. PSO with a constriction (CPSO) defined a “no-hope” convergence criterion and a “rehope” method as well as one social/confidence parameter to re-initialize the swarm [15,16]. Attractive and repulsive PSO (ARPSO) alternated between phases of attraction and repulsion according to the diversity value of the swarm, which prevented premature convergence to a high degree at a rapid convergence like adaptive PSO, [17]. Two improved ARPSOs introducing a mixed phase combined with backpropagation (BP) were proposed in [18], which had good performance on function approximation and benchmark classification problems. In fuzzy adaptive PSO (FAPSO), a fuzzy system was implemented to dynamically adapt the inertia weight of PSO, which was especially useful for optimization problems with a dynamic environment [19]. To obtain better approximation performance, double search methods, which combined adaptive PSO encoding prior constraints from the approximated functions with gradient-descent-based algorithm, were proposed [20,21]. These double search methods were designed for function approximation using FNN, and they increased computational complexity.

Although random search such as adaptive PSO converges faster than GA, it is easy to converge to local minima because of losing

* Corresponding author. Tel.: +86 511 88790321 533; fax: +86 511 88780371.
E-mail addresses: hanfei@ujs.edu.cn (F. Han), qingjing_03@126.com (Q. Liu).

http://dx.doi.org/10.1016/j.neucom.2013.03.074
0925-2312

the diversity of the swarm. On the other hand, some improved PSOs, such as PSOPC, CPSO, fussy PSO and ARPSO, may improve their search ability, but they require more time to find the best solution.

In this paper, an improved hybrid particle swarm optimization termed DGHPSOGS is proposed to overcome the problem of premature convergence. The improved method takes advantages of random search and semi-deterministic search. When the diversity value of the swarm is below a predetermined threshold, the proposed diversity-guided PSO based on gradient search called DGPSOGS is used to perform search. On the other hand, the particles update their positions by the adaptive PSO (APSO) as the diversity value of the swarm is greater than the threshold. In the hybrid method, the APSO is used to perform stochastic search by attracting the particles and the DGPSOGS is used to perform semi-deterministic search as well as repel the particles to keep the diversity of the swarm in a reasonable range. Therefore, the hybrid algorithm can decrease the likelihood of premature convergence with a reasonable convergence rate.

The rest of the paper is organized as follows. Section 2 introduces some preliminaries of PSO. The proposed method is proposed in Section 3. In Section 4, experimental results and discussion for optimization problems are given to demonstrate the efficiency and effectiveness of the proposed algorithm. Finally, the concluding remarks are offered in Section 5.

2. Particle swarm optimization

2.1. Basic particle swarm optimization

PSO is an evolutionary computation technique that searches for the best solution by simulating the movement of birds in a flock [8,9]. The population of the birds is called swarm, and the members of the population are particles. Each particle represents a possible solution to the optimizing problem. During each iteration, each particle flies independently in its own direction, which is guided by its own previous best position as well as the global best position of all the particles. Assume that the dimension of the search space is D, and the swarm is $S=(X_1, X_2, X_3, \ldots, X_{Np})$; each particle represents a position in the D dimension; the position of the i-th particle in the search space can be denoted as $X_i=(x_{i1}, x_{i2}, \ldots, x_{iD})$, $i=1, 2, \ldots, N_p$, where N_p is the number of all particles. The own previous best position of the i-th particle is called *pbest* which is expressed as $P_i=(p_{i1}, p_{i2,} \ldots, p_{iD})$. The best position of all the particles is called *gbest,* which is denoted as $P_g=(p_{g1}, p_{g2,} \ldots, p_{gD})$. The velocity of the i-th particle is expressed as $V_i=(v_{i1}, v_{i2}, \ldots, v_{iD})$. According to [8,9], the basic PSO is described as

$$V_i(t+1)=V_i(t)+c1\times rand()\times(P_i(t)-X_i(t)) \\ +c2\times rand()\times(P_g(t)-X_i(t)) \tag{1}$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \tag{2}$$

where $c1$ and $c2$ are the acceleration constants with positive values; and $rand()$ is a random number ranging from 0 to 1.

To obtain better performance, adaptive particle swarm optimization (APSO) algorithm was proposed [22], and the corresponding velocity update of particles was denoted as follows:

$$V_i(t+1)=W(t)\times V_i(t)+c1\times rand()\times(P_i(t)-X_i(t))+c2 \\ \times rand()\times(P_g(t)-X_i(t)) \tag{3}$$

where the inertia weight $W(t)$ can be computed by the following equation:

$$W(t)=W_{\max}-t\times(W_{\max}-W_{\min})/N_{pso} \tag{4}$$

In Eq. (4), W_{max}, W_{min} and N_{pso} are the initial inertial weight, the final inertial weight and the maximum iterations, respectively.

2.2. Attractive and repulsive particle swarm optimization (ARPSO)

In [17], attractive and repulsive particle swarm optimization (ARPSO), a diversity-guided method, was proposed which was described as

$$V_i(t+1)=W(t)\times V_i(t)+dir[c1\times rand()\times(P_i(t)-X_i(t)) \\ +c2\times rand()\times(P_g(t)-X_i(t))] \tag{5}$$

where $dir=\begin{cases}-1 & diversity<d_{low} \\ 1 & diversity>d_{high}\end{cases}$.

In ARPSO, a function was proposed to calculate the diversity of the swarm as follows:

$$diversity\ ()=\frac{1}{N_p\times|L|}\times\sum_{i=1}^{N_p}\sqrt{\sum_{j=1}^{D}(p_{ij}-\overline{p_j})^2} \tag{6}$$

where $|L|$ is the length of the maximum radius of the search space; p_{ij} is the j-th component of the i-th particle and $\overline{p_j}$ is the j-th component of the average over all particles.

In the attraction phase ($dir=1$), the swarm is attracting, and consequently the diversity decreases. When the diversity drops below the lower bound, d_{low}, the swarm switches to the repulsion phase ($dir=-1$). When the diversity reaches the upper bound, d_{high}, the swarm switches back to the attraction phase. ARPSO alternates between phases of exploiting and exploring – attraction and repulsion – low diversity and high diversity and thus improves its search ability [17].

3. The proposed hybrid algorithm

3.1. The diversity-guided PSO based on gradient search (DGPSOGS)

Since the negative gradient always points in the direction of the steepest decrease in the cost function, the nearest local minimum (maybe global minimum) will be reached eventually. In the new PSO, the particles fly to the global minimum according to their gradient of the fitness function. In the case of a function with a single minimum (unimodal function) the gradient descent algorithm converges faster than stochastic search algorithms because stochastic search algorithms are a waste of computational effort doing a random search. For multimodal functions, the gradient descent algorithm will converge to the local minimum closest to the starting point [13]. When the swarm is trapped in the local minimum, the swarm will lose its diversity. To make the swarm jump out of the local minimum, the particles in the swarm should be repelled by each other, thus improving the diversity of the swarm. For quickly finding the global minimum as well as avoiding being trapped into local minimum, the particle velocity update equation in DGPSOGS is proposed as follows:

$$V_i(t+1)=W(t)\times V_i(t)+c1\times rand() \\ \times\left(\frac{-\partial f(X_i(t))/\partial X_i(t)}{\|-\partial f(X_i(t))/\partial X_i(t)\|}\right)-c2\times rand()\times\frac{(P_g(t)-X_i(t))}{diversity(t)+\xi} \tag{7}$$

where $f()$ is the fitness function; and ξ is a predetermined small positive number for fear that the denominator is equal to zero.

The second term on the right side of Eq. (7) guides the particles to search in the negative gradient direction. The third term on the right side of Eq. (7) is inversely proportional to the diversity of the swarm, $diversity(t)$. The smaller the diversity of the swarm, the greater the value of the third term, which results in the particles repelling each other more greatly and improving the diversity of the swarm. Although the DGPSOGS uses gradient information to accurately find the local minima, it can jump out of the local minima by improving the diversity of the swarm. Therefore, Eq. (7) ensures the swarm not only to search in the deterministic direction but also not to lose its diversity, which increases the likelihood of the swarm converging to the global minima.

3.2. The proposed hybrid method (DGHPSOGS)

The proposed hybrid method combines random search and semi-deterministic search to improve its search ability. The APSO is first used to perform random search till the diversity value of the swarm is less than a predetermined threshold, d. When the swarm loses its diversity, the DGPSOGS algorithm is used to perform semi-deterministic search. The flowchart of the proposed hybrid method is shown in Fig. 1.

In the hybrid method, whether the APSO or DGPSOGS is used to perform search is decided by the diversity value of the swarm. The search alternates between the APSO and DGPSOGS. The APSO may cause the swarm to lose its diversity, and the swarm is trapped into local minima with high likelihood. On the other hand, the DGPSOGS can keep the diversity of the swarm adaptively as well as search in the negative gradient direction, which may increase the likelihood of finding the global minima at a faster convergence rate than other diversity-guided PSOs.

The computational complexity of the hybrid method is the sum of the APSO and DGPSOGS. Since the computational complexity of the DGPSOGS has the same order of magnitude as the one of the APSO, the hybrid method does not increase the computational complexity compared with APSO. Since the proposed method is a diversity guided hybrid PSO with gradient search, it is referred to as DGHPSOGS.

4. Experimental results

In this section, the performance of the proposed hybrid PSO algorithm is compared to that of the APSO, CPSO, PSOPC, FAPSO and ARPSO on some functions in the De Jong test suite of benchmark optimization problems. The problems provide common challenges that an evolutionary computation algorithm is expected to face, such as multiple local minima and flat regions

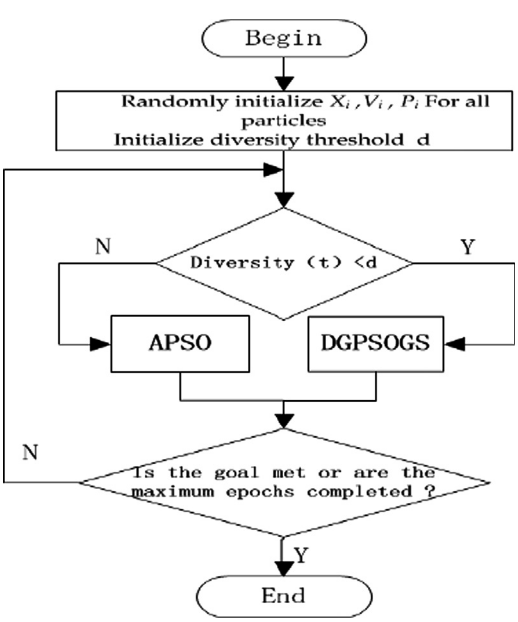

Fig. 1. The flowchart of the DGHPSOGS.

Table 1
Five test functions used for comparing DGHPSOGS to other PSOs.

Test function	Equation	Search space
Sphere (F1)	$\sum_{i=1}^{n} x_i^2$	$(-100,\ 100)^n$
Ellipsoid (F2)	$\sum_{i=1}^{n} i x_i^2$	$(-100,\ 100)^n$
Rosenbrock (F3)	$\sum_{i=1}^{n-1} (100(x_{i+1}-x_i^2)^2+(1-x_i)^2)$	$(-100,\ 100)^n$
Rastrigin (F4)	$10n+\sum_{i=1}^{n} (x_i^2-10\cos(2\pi x_i))$	$(-100, 100)^n$
Griewangk (F5)	$1+\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i-1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$(-100, 100)^n$

surrounding the global minimum. Table 1 shows some classic test functions used in the experiments. The Sphere and Ellipsoid functions both are convex and unimodal (single local minimum). The Rosenbrock test function has a single global minimum located in a long narrow parabolic-shaped flat valley and tests the ability of an optimization algorithm to navigate flat regions with small gradients. The Rastrigin and Griewangk functions are highly multimodal and test the ability of an optimization algorithm to escape from local minima.

The population size for PSOPC is 120 and the ones for the other five PSOs are all 20 in all experiments. Moreover, the acceleration constants $c1$ and $c2$ for PSO, FAPSO and ARPSO are all set as 2.0. The constants $c1$ and $c2$ are both 2.05 in CPSO and 0.5 in PSOPC. In the proposed hybrid algorithm, $c1$ and $c2$ are both 2.1. The decaying inertia weight w starting at 0.9 and ending at 0.4 is set for APSO, CPSO, ARPSO and DGPSOGS according to [22]. The initial inertial weight and the final one in PSOPC are 0.9 and 0.7, respectively. The parameter *CBPEmin* and *CBPEmax* in FAPSO are set as 0 and 1000 for the functions F1 and F2. As for the functions F3, F4 and F5, the selection of the *CBPEmin* and *CBPEmax* is the same as that in [19]. All the results shown in this paper are the mean values of 20 trials.

Without loss of generality, Fig. 2 shows the mean convergence curve of different PSOs on all five functions with ten dimensions. Fig. 2(a) and (b) shows that the gradient-descent-based algorithm (DGHPSOGS) performs better than stochastic search algorithms (APSO, CPSO, FAPSO and ARPSO) on two unimodal and convex functions. For the function F1, PSOPC has a better search performance than DGHPSOGS, whereas DGHPSOGS performs better than PSOPC for the function F2. CPSO, FAPSO and APSO converge to local minima around 100, 600 and 1200 iterations, respectively, for both the functions. Although ARPSO does not converge to local minimum, it also does not converge to global minimum at 3000 iterations for the two functions. The DGHPSOGS converges to global minimum at nearly 1200 and 1600 iterations for the Sphere and Ellipsoid test functions, respectively. Moreover, the DGHPSOGS converges much more accurately than all other PSOs for the two functions but PSOPC for function F1. From Fig. 2(c), APSO, CPSO and ARPSO lose global search ability early because the convergence accuracies of CPSO, APSO, FAPSO, PSOPC and ARPSO have not been changed nearly after 300, 3000, 2000, 2300 and 9400 iterations, which indicates that the swarm is trapped in local minimum. The convergence accuracy of the DGHPSOGS has been decreasing continuously during the whole search process and the proposed hybrid PSO converges most accurately in all PSOs. From Fig. 2(d) and (e), the DGHPSOGS converges significantly faster to a better solution than other PSOs on the multimodal Rastrigin and Griewangk test functions.

Table 2 shows the mean best solution for the five test functions using six PSOs. From Table 2, the proposed algorithm is found to have better convergence accuracy than other PSOs in most cases. Although PSOPC obtains the best convergence accuracy on the Sphere function with 10 dimensions and on the Rastrigin function with 20 and 30 dimensions, it requires more particles than other

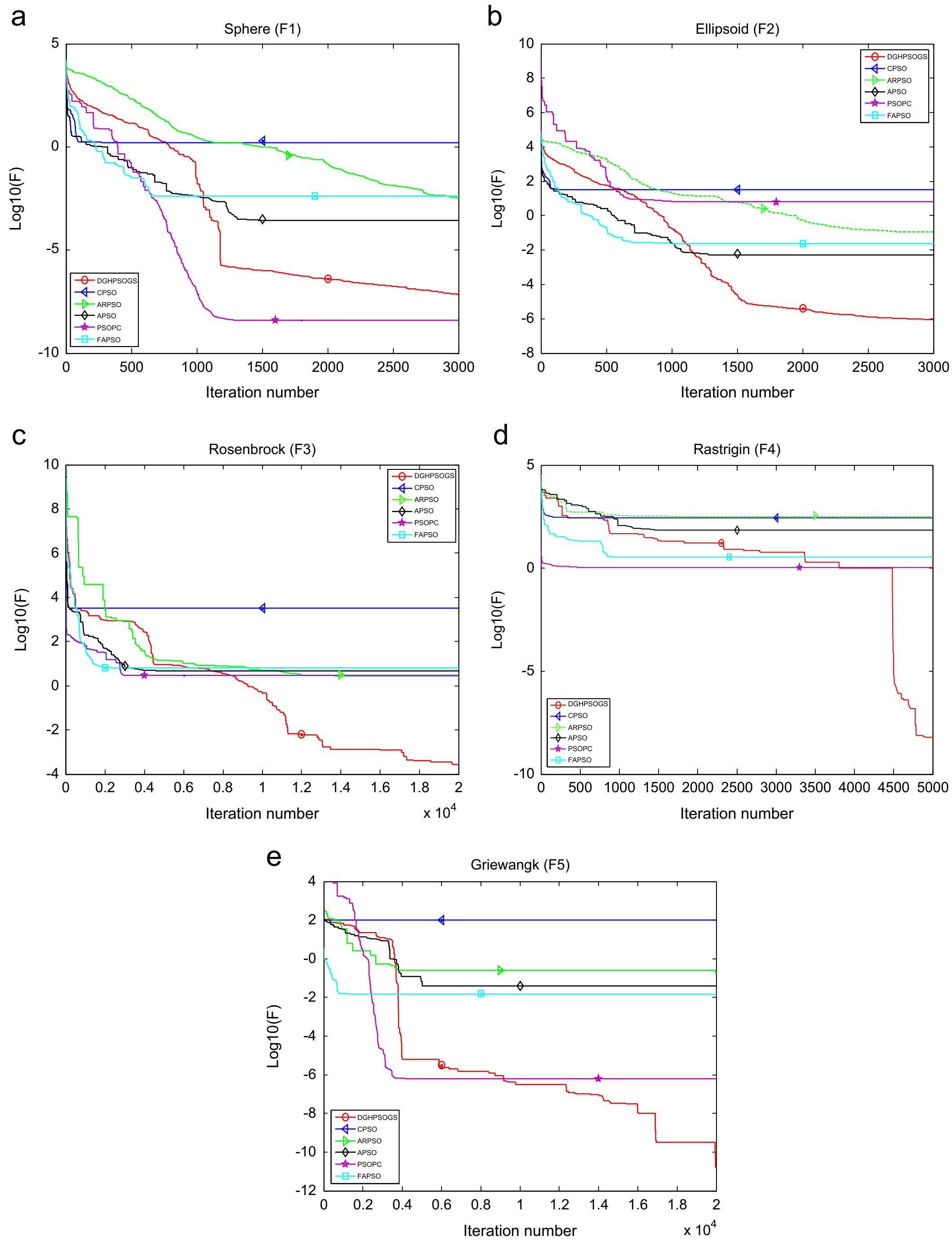

Fig. 2. Mean best solution versus iteration number for the five test functions ($n=10$) using six PSOs: (a) Sphere, (b) Ellipsoid, (c) Rosenbrock, (d) Rastrigin, and (e) Griewangk.

Table 2
Mean best solution for the five test functions using six PSOs.

Test function	Dimension	Mean best solution					
		APSO	CPSO	PSOPC	FAPSO	ARPSO	DGHPSOGS
Sphere (F1)	10	2.80e−4	1.64	5.89e−9	1.1e+3	3.3e−3	1.26e−8
	20	0.22	1.11e+3	8.42e−7	0.93	0.38	2.52e−7
	30	0.21	4.43e+3	5.33e−7	9.04	11.02	2.64e−7
Ellipsoid (F2)	10	5.1e−3	33.42	6.97	1.15e+2	0.11	3.37e−7
	20	2.91	5.94e+3	37.89	25.00	66.72	7.93e−6
	30	135.60	3.33e+4	432.30	121.46	6.11e+3	1.58e−5
Rosenbrock (F3)	10	4.73	1.46e+4	7.98	6.54	2.73	4.50e−4
	20	21.51	1.59e+5	18.63	52.10	191.19	5.16e−4
	30	65.06	1.30e+6	60.31	220.31	352.68	5.5e−3

Table 2 (*continued*)

Test function	Dimension	Mean best solution					
		APSO	CPSO	PSOPC	FAPSO	ARPSO	DGHPSOGS
Rastrigin (F4)	10	2.7e−2	107.54	0.058	3.44	79.48	9.37e−6
	20	3.29	433.15	0.42	39.78	14.91	2.95
	30	4.11	3.18e+3	1.09	78.44	40.58	25.57
Griewangk (F5)	10	7.6e−2	0.36	5.32e−7	0.093	0.14	3.03e−11
	20	0.34	1.10	7.50e−5	0.0192	0. 67	4.30e−9
	30	1.68	1.38	7.21e−3	0.0124	1.11	6.01e−8

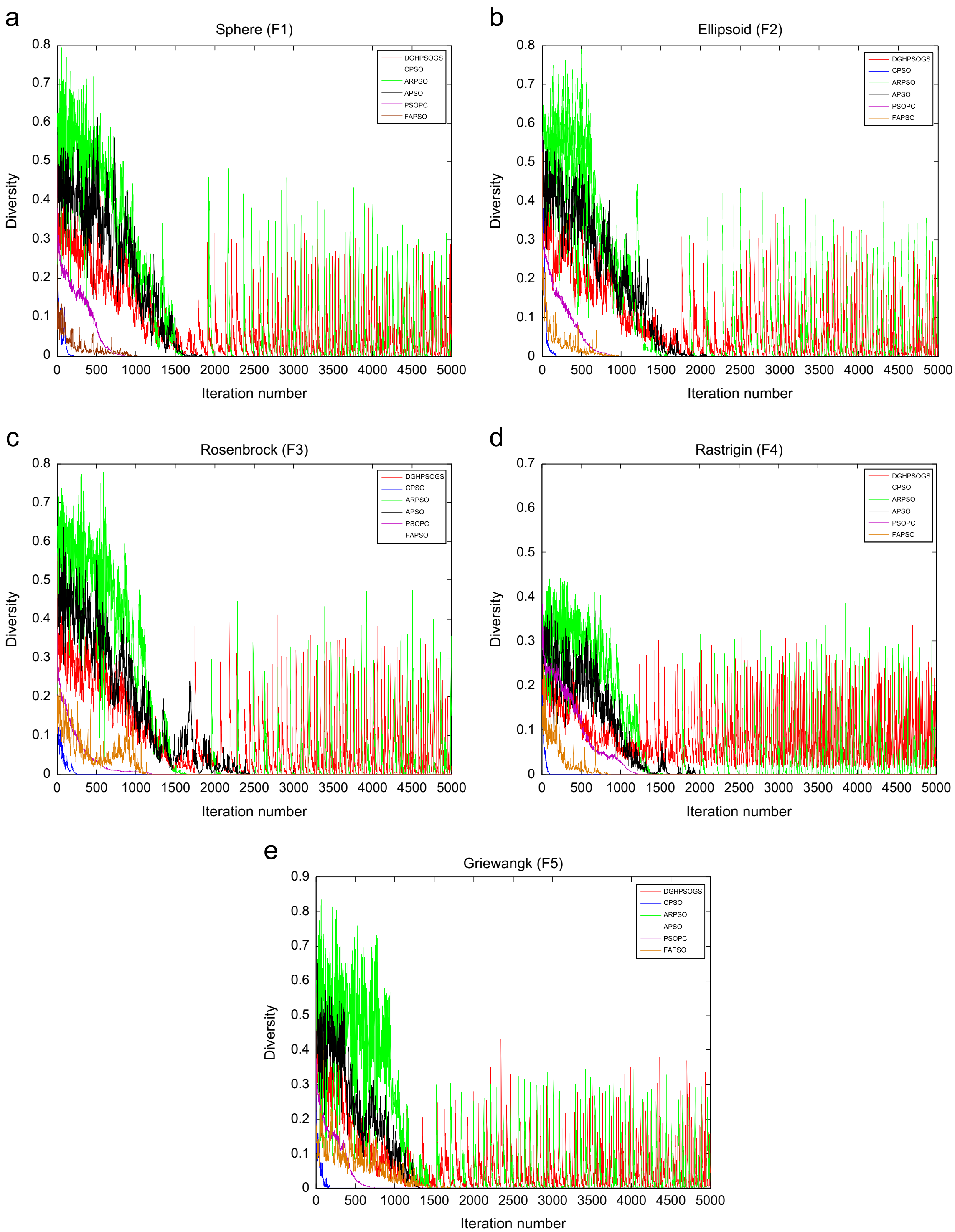

Fig. 3. The diversity values of six PSOs on five test functions: (a) Sphere, (b) Ellipsoid, (c) Rosenbrock, (d) Rastrigin and (e) Griewangk.

PSOs. In general, the proposed hybrid PSO is superior to PSOPC and other PSOs.

Without loss of generality, the following experiments are conducted on the test functions with ten dimensions. Fig. 3 shows the diversity value of the swarm in six PSOs for the five test functions. Obviously, the swarms in the APSO, CPSO, FAPSO and PSOPC lose their diversity quickly, while the ARPSO and DGHPSOGS keep the diversity of the swarm adaptively in the whole search process. Moreover, the DGHPSOGS adjusts the diversity values of the swarm greater than the predetermined value with much less iteration than the ARPSO when the swarm loses its diversity, which indicates the DGHPSOGS controls the diversity of the swarm more efficiently than the ARPSO.

Fig. 4 shows the iteration distribution in different diversity intervals of the swarm for different PSOs. In the APSO, CPSO, FAPSO and PSOPC, most of the iterations are in the low diversity interval (less than 1.0e−6), which shows that these PSOs easily lose the diversity of the swarm and cannot improve the diversity effectively. The iterations are distributed in all intervals comparatively equally in the ARPSO, while the iterations are distributed mainly in two or three intervals ((1.0e−3,1.0e−2), (1.0e−2, 1.0e−1) and (1.0e−1,1)) in the proposed PSO. This also illustrates that the ARPSO and DGHPSOGS both can adaptively adjust the diversity of the swarm and the latter adjusts the diversity more quickly than the former.

Below, we discuss the effects of the parameters in the DGHPSOGS. Fig. 5 shows the relationship between the predetermined threshold (d) and the best solution for the five test functions using the proposed hybrid algorithm. The proposed algorithm is not sensitive to the threshold on functions F1 and F2. For function F3, the greater the threshold, the worse the search performance of the DGHPSOGS. The suitable threshold values in the proposed algorithms are $10^{-2.5}$, 10^{-4}, $10^{-5.7}$, $10^{-2.8}$ and $10^{-2.1}$ for Sphere, Ellipsoid, Rosenbrock, Rastrigin and Griewangk, respectively.

Fig. 6 shows the relationship between the acceleration constants ($c1$ and $c2$) in the DGHPSOGS and the best solutions for the five test functions. Case I: $c2=2.1$, 1.8, 2.2, 2.0 and 2.3 is kept unchanged for F1, F2, F3, F4 and F5, respectively, $c1$ is selected from [1,3] at an interval of 0.2. Case II: $c1=2.0$, 2.1, 2.3, 2.1 and 2.5 is kept unchanged for F1, F2, F3, F4 and F5, respectively, $c2$ is selected from [1,3] at an interval of 0.2. The corresponding results are listed in Fig. 6(a) and (b). It can be found that two suitable intervals for the selection of $c1$ and $c2$ are [2, 2.5] and [1.8, 2.5], respectively.

5. Conclusions

To improve the search ability, a hybrid PSO combining random search and semi-deterministic search was proposed in this paper. In the hybrid PSO, the search alternated from the APSO and DGPSOGS according to the diversity value of the swarm. The APSO led the swarm to converge to local minima. The DGPSOGS kept the diversity of the swarm adaptively as well as search in the negative gradient direction, which increased the likelihood of finding the global minima at a faster convergence rate. The experimental results were provided to verify that the proposed hybrid

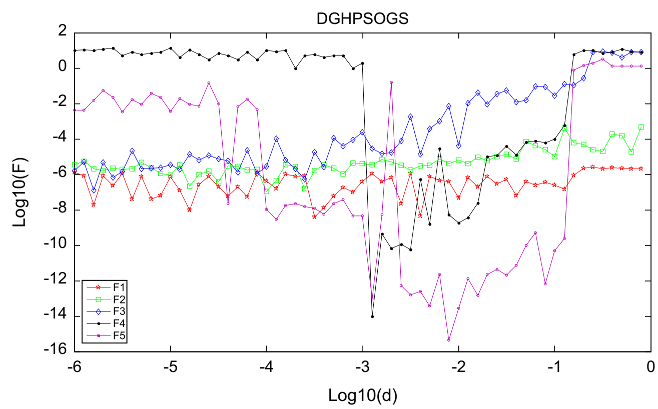

Fig. 5. The predetermined diversity threshold (d) for DGHPSOGS versus the best solutions for the five test functions.

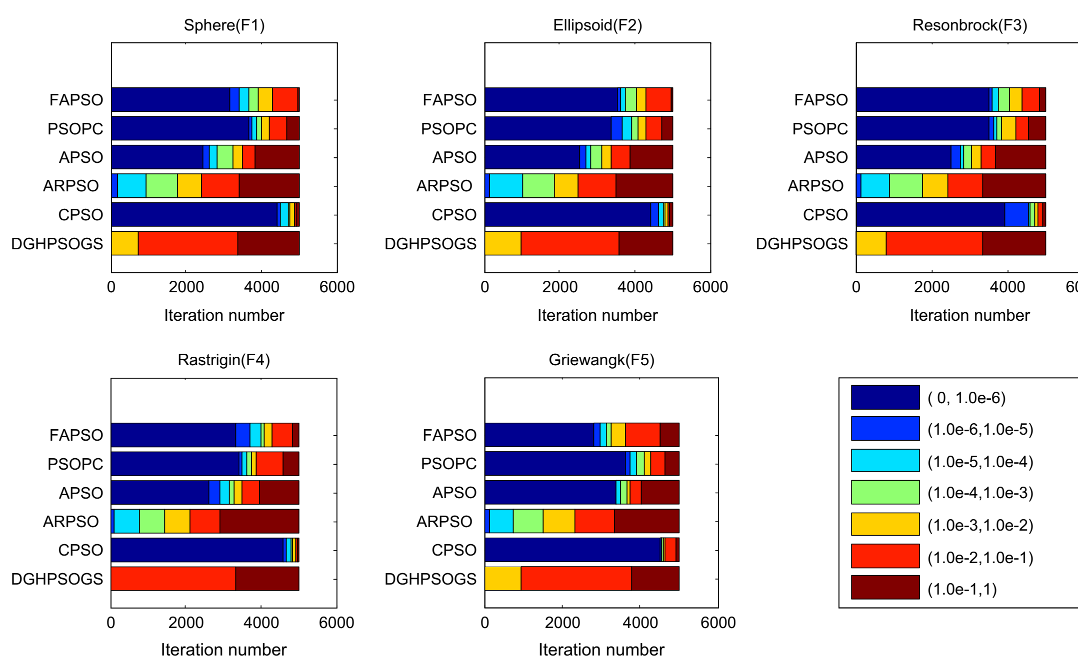

Fig. 4. The iteration distribution in different intervals for different PSOs.

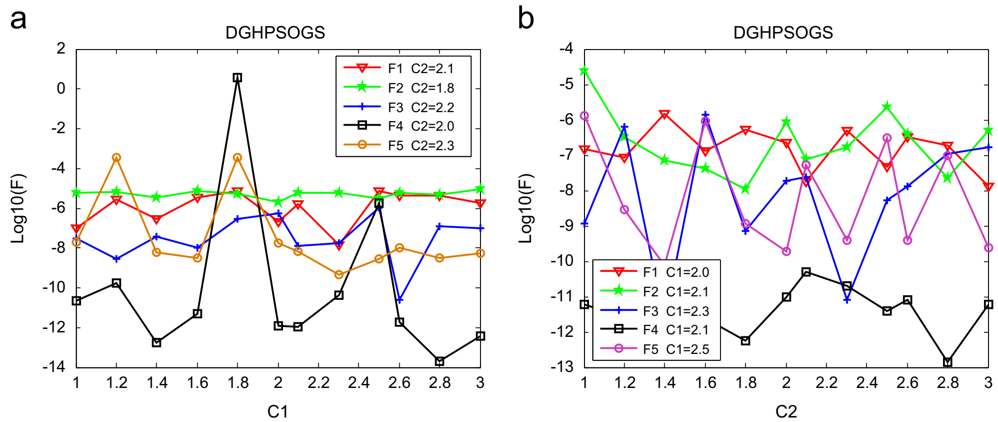

Fig. 6. The acceleration constants ($c1$ and $c2$) for DGHPSOGS versus the best solutions for the five test functions: (a) Case I and (b) Case II.

algorithms had better convergence performance with a faster convergence rate than APSO, CPSO, PSOPC, FAPSO and ARPSO. In the future research works, we will use this promising hybrid algorithm for more complex problems.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61271385 and 60702056) and the Initial Foundation of Science Research of Jiangsu University (No. 07JDG033).

References

[1] D.S. Huang, The local minima free condition of feedforward neural networks for outer-supervised learning, IEEE Trans. Syst., Man Cybern. 28 (3) (1998) 477–480.
[2] D.S. Huang, Radial basis probabilistic neural networks, model and application, Int. J. Pattern Recognit. Artif. Intell. 13 (7) (1999) 1083–1101.
[3] D.S. Huang, Ji-Xiang Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE Trans. Neural Netw. 19 (12) (2008) 2099–2115.
[4] W.H. Joerding, J.L. Mcador, Encoding a prior information in feedforward networks, Neural Netw. 4 (1991) 847–856.
[5] D.S. Huang, A constructive approach for finding arbitrary roots of polynomials by neural networks, IEEE Trans. Neural Netw. 15 (2004) 477–491.
[6] D.S. Huang, Zheru Chi, Finding roots of arbitrary high order polynomials based on neural network recursive partitioning method, Sci. China Ser. F Inf. Sci. 47 (2004) 232–245.
[7] D.S. Huang, Horace H.S. Ip, Zheru Chi, A Neural root finder of polynomials based on root momnets, Neural Comput. 16 (2004) 1721–1762.
[8] J. Kennedy, R.C. Eberhart, Particle swarm optimization, IEEE Int. Conf. Neural Netw. 4 (1995) 1942–1948.
[9] R.C. Eberhart J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machines and Human Science, 1995, pp. 39–43.
[10] Crina Grosan, Ajith Abraham, A novel global optimization technique for high dimensional functions, Int. J. Intelligent Syst. 24 (4) (2009) 421–440.
[11] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[12] Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput. 12 (2008) 171–195.
[13] Mathew M. Noel, A new gradient based particle swarm optimization algorithm for accurate computation of global minimum, Appl. Soft Comput. 12 (1) (2012) 353–359.
[14] S. He, Q.H. Wu, J.Y. Wen, A particle swarm optimizer with passive congregation, Biosystems 78 (2004) 135–147.
[15] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, Piscataway, NJ: IEEE Service Center, 1999, pp. 1951–1957.
[16] D. Corne, M. Dorigo, F. Glover, New Ideas in Optimization, McGraw Hill, London, UK (1999) 379–387 (chapter 25).
[17] J. Riget, J.S. Vesterstrom, A diversity-guided particle swarm optimizer – the arPSO, Technical Report 2, 2002.
[18] Fei Han, Jiansheng Zhu, Improved particle swarm optimization combined with backpropagation for feedforward neural networks, Int. J. Intelligent Syst. 28 (3) (2013) 271–288.
[19] Y. Shi, R.C. Eberhary, Fuzzy adaptive particle swarm optimization, Evol. Comput. 1 (2001) 101–106.
[20] Fei Han, Qing-Hua Ling, A new approach for function approximation incorporating adaptive particle swarm optimization and a priori information, Appl. Math. Comput. 205 (2008) 792–798.
[21] Fei Han, Qing-Hua Ling, De-Shuang Huang, An improved approximation approach incorporating particle swarm optimization and a priori information into neural networks, Neural Comput. Appl. 19 (2) (2008) 255–261.
[22] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, Comput. Intell. 6 (1998) 69–73.

Fei Han received his M.A. degree from Hefei University of Technology in 2003 and his Ph.D. degree from University of Science and Technology of China in 2006. He is currently an Associate Professor of computer science at Jiangsu University. His principal research interests are intelligent computing and intelligent information processing, including neural networks, particle swarm optimization, and bioinformatics.

Qing Liu is a graduate student in the School of Computer Science and Telecommunication Engineering at Jiangsu University. He received his B.S. degree from Inner Mongolia University in 2009. His research interests include particle swarm optimization and intelligent computing.