Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 1437–1443
Progress in Natural Science
www.elsevier.com/locate/pnsc

Particle swarm optimization with a leader and followers

Junwei Wang *, Dingwei Wang

Institute of Systems Engineering, College of Information Science and Engineering, Northeastern University, Shenyang 110004, China

Received 4 March 2008; accepted 31 March 2008

Abstract

Referring to the flight mechanism of wild goose flock, we propose a novel version of Particle Swarm Optimization (PSO) with a leader and followers. It is referred to as Goose Team Optimization (GTO). The basic features of goose team flight such as goose role division, parallel principle, aggregate principle and separate principle are implemented in the recommended algorithm. In GTO, a team is formed by the particles with a leader and some followers. The role of the leader is to determine the search direction. The followers decide their flying modes according to their distances to the leader individually. Thus, a wide area can be explored and the particle collision can be really avoided. When GTO is applied to four benchmark examples of complex nonlinear functions, it has a better computation performance than the standard PSO.

Keywords: Particle swarm optimization; Goose team optimization; Role division; Parallel principle; Aggregate principle; Separate principle

1. Introduction

Learning from life system, people have developed many optimization computation methods to solve complicated problems in the recent decades. Genetic algorithm [1], artificial immune systems [2], artificial neural network [3], ant colony optimization [4], culture algorithm [5], colony location algorithm [6] have been widely used in many industrial and social areas. We call this kind of algorithms for scientific computation as "Artificial-Life Computation" [6]. Particle swarm optimization (PSO) is such a new computation technique developed by Kennedy and Eberhart based on the simulation of a simplified social model [7–9]. It has been attracting more and more attention for its simple concept and easy implementation with only a few parameters.

The underlying motivation for the development of the PSO algorithm is the social behavior of animals such as bird flocking, fish schooling and swarm theory [10]. Whereas there are different patterns in the animal aggregations in the nature, for example, goose team,is just a kind of special group. In a goose team, there is the division of goose roles and the whole team is divided into two parts: a leader and the followers. A follower flies in the same direction with the leader, and the distance between them should not be too long or too short, which is very different from other bird flocks. In this paper, particle swarm optimization with leader and followers is proposed by introducing the role division, parallel principle, aggregate principle and separate principle into the standard PSO model. In such an algorithm, the individuals work more like a team than like a swarm, so we can call this model as "Goose Team Optimization (GTO)". The leader guides the followers flying, and the parallel principle enables the goose team to search in the same direction. The aggregate principle makes the followers close to the leader. The separate principle avoids the goose collision in the team and keeps the diversity of the population.

This paper analyzes the difference between the particle swarm and the goose team, and sums up the principles of

* Corresponding author. Tel.: +86 24 83682292; fax: +86 24 23893138.
E-mail address: wangjunwei@ise.neu.edu.cn (J. Wang).

1002-0071/$ - see front matter © 2008 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.
doi:10.1016/j.pnsc.2008.03.029

the goose team which are added into the standard PSO model. The basic ideas and the procedures of particle swarm optimization with leader and followers are described. Four benchmark examples of complex nonlinear functions are used to test the algorithm, and a better performance has been achieved.

2. Standard particle swarm optimization

The particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart based on the simulation of a simplified social model [10–12]. The algorithm has attracted more and more attention now [13,14]. The standard PSO (SPSO) algorithm can be expressed as follows.

A swarm making up by m particles searches a D-dimensional problem space. Each particle is assigned a randomized velocity and a stochastic position. The position represents the solution of the problem. When flying each particle is attracted to the good location achieved so far by itself and to the good location achieved by the members in the whole swarm (or by the members in the neighborhood). The position of the i th particle is represented as $x_i = (x_{i1},\ldots,x_{id},\ldots,x_{iD})$ and its velocity is represented as $v_i = (v_{i1},\ldots,v_{id},\ldots,v_{iD})$, $1 \leqslant \mathrm{i} \leqslant \mathrm{m}$, $1 \leqslant \mathrm{d} \leqslant \mathrm{D}$,. The best previous position of the i th particle, namely the position with the best fitness value, is represented as $p_i = (p_{i1},\ldots, p_{id},\ldots,p_{iD})$, and the index of the best particle among all the particles in the population (or in the neighborhood) is represented by the symbol g. Each particle updates its velocity and position according to the following equations:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1\xi(p_{id}^k - \chi_{id}^k) + c_2\eta(p_{gd}^k - \chi_{id}^k) \quad (1)$$

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

where ω is the inertia weight that determines how much a particle holds its current velocity in the next iteration. Suitable selection of inertia weight can provide the particles with a balance between the ability of exploitation and exploration. c_1 and c_2 are learning factors, also named acceleration constants, which are two positive constants. Learning factors are usually equal to 2, while other settings can also be seen in some papers. ξ, $\eta \in U[0,1]$, and they are pseudo-random numbers that obey the same homogeneous distribution in the range [0,1]. The velocity of a particle is limited in the range of $V_{\max}$. There have been tests and analyses that indicate that the effect of setting $V_{\max}$ can be replaced by the tuning of inertia weight. $V_{\max}$ is set to be the range of each dimension variable and is used to initialize the velocity of particles without selecting and tuning in detail in the experiments. The implementing steps of PSO are just like other evolutionary algorithms, such as GA, which include initialization, fitness evaluation, update of velocity and position, and testing of stop criterion. We do not describe then in detail here.

3. Particle swarm optimization with leader and followers

3.1. Analysis of goose team flight

The aggregate motion of animals, such as a flock of birds, a herd of land animals, or a school of fish, is a beautiful and a familiar part of the nature world [13]. The basic urge to join a flock seems to be the result of evolutionary pressure from several factors: protection from predators, statistically improving survival of the gene pool from attacks from predators, profiting from a large effective search pattern in the quest for food, and advantages for social and mating activities [14]. The pattern and complexity of the animal aggregation are interesting problems which attract not only many biologists, but also some scientists on artificial intelligence. Reynolds [13] concluded that the behaviors that led to simulated flocking were

1. Collision avoidance: avoid collision with nearby flockmates.
2. Velocity matching: attempt to match velocity with nearby flockmates.
3. Flock centering: attempt to stay close to nearby flockmates.

In such a flock, there is no role division and there is no leader. Particle swam optimization algorithm is just based on the simulation of bird flock and fish school. The term *swarm* is used in accordance with a paper by Millonas [10]. Particles in a swarm do not have the behavior of collision avoidance and velocity matching, and there is no difference between them.

Goose team is a special bird flock, which flies in a V-formation. Anderson and Franks [15] suggested that a team task requires different subtasks to be performed concurrently for success completion, and a team is simply the set of individuals who perform a team task. There is a division of labor within a team. Goose team is such an example. Goose team flies in a V-formation during the migration and there are two subtasks within the team, leading and following. Each bird takes advantages of the uplifting air from the bird in front of it. This updraft actually lifts the bird, making the flight a little easier. As each goose flaps its wings, it creates a lift for the bird that follows. When a goose falls out of the formation, it feels the drag and the resistance of flying alone. It quickly moves back into the formation to take advantage of the lifting power of the bird immediately in front of it. When the leader is tired, it rotates back into the formation, and another goose flies to the point position. By flying in a V-formation, the whole team of 25 birds adds 71 percent extra flying range [16].

As analysed above, the flight mechanism of a goose team can be described as follows:

1. Role division: geese in a team can be divided into two roles: leader and follower. And there are different subtasks for different roles. The division of roles is dynamic, and there is a periodic update of the leader.
2. Separate principle: the distance between a follower and the leader should not be too short. Too short distance will result in collision.
3. Aggregate principle: the distance between a follower and the leader should not be too long. A goose cannot enjoy the uplifting air from the bird in front of it at a long distance.
4. Parallel principle: a follower should be in accordance with the leader in the flight direction when there is a proper distance to the leader.

3.2. Basic idea of GTO

Benefiting from the flight mechanism of goose team, we design a novel version of particle swarm optimization with leader and followers to optimize the complicated nonlinear functions. It is referred to as "Goose Team Optimization". The basic optimization ideas can be described as follows:

1. Role division: a goose team is divided into two different parts, a leader and the followers, based on the different subtasks. In each iteration, the goose getting best fitness is set as the leader, which determines a good search direction for the team.
2. Separate principle: to prevent excessive aggregation of the goose team, we set a relative short distance as threshold A. The follower should fly away from the leader when the distance between a follower and the leader is shorter than A. The detailed implementation of the separate principle is as follows.

Let the initial space of the function be $[-l,l]^n$, and the distance between two points in the space is expressed as Euclid distance. The maximal distance between two points in the initial space is $2l\sqrt{n}$. Set a relative low positive decimal fraction $\lambda_{\min}$, $0 < \lambda_{\min} < 1$, and the threshold A is expressed as $A = 2\lambda_{min} l\sqrt{n}$. Here, we call $\lambda_{\min}$ as the shortest distant coefficient. It is obvious that the threshold A should be adjusted according to the problem.

We denote the distance between a follower i and the leader l as d_{il}. When $d_{il} < A$, the follower updates its position as Eq. (3).

$$\chi_{id}^{k+1} = \chi_{id}^{k} + c_1\xi(\chi_{id}^{k} - \chi_{id}^{k}) \tag{3}$$

where c_1 is a positive constant which provides the follower with the ability to move away from the leader, and prevents the excessive aggregation of the team. c_1 can be equal to 2. $\xi \epsilon U[0,1]$, and it is a pseudo-random number obeying the homogeneous distribution in the interval [0,1].

3. Aggregate principle: the distance between a follower and the leader should not be too long. We set a relative long distance as threshold B. The follower should fly close to the leader, when the distance between a follower and the leader is longer than the threshold B. The detailed implementation of the aggregate principle is as follows:

Set a relative high positive decimal fraction $\lambda_{\max}$, $0 < \lambda_{\max} < 1$, and the threshold B is expressed as $B\lambda_{max} l\sqrt{n}$. Here, we call $\lambda_{\max}$ as the longest distant coefficient. The threshold B also needs adjustment according to the problem.

When $d_{il} > B$, the follower updates its position as Eq. (4).

$$\chi_{id}^{k+1} = \chi_{id}^{k} + c_2\eta(\chi_{id}^{k} - \chi_{id}^{k}) \tag{4}$$

where c_2 is a positive constant which provides the follower with the ability to move close to the leader. c_2 can be equal to 2. $\eta \in U[0,1]$, and it is a pseudo-random number obeying the homogeneous distribution in the interval [0,1].

4. Parallel principle: the follower flies in the same direction with the leader, when the distance between a follower and the leader is between the threshold A and the threshold B. In the optimization process of the complicated nonlinear functions the leader uses the negative gradient direction as the search direction. The characteristic of the function itself is taken into account by using the gradient information. And the points without gradient are separate in a continuous function, and they will not affect the optimization result.

3.2.1. Implementation of GTO

The procedure of goose team optimization for continuous nonlinear functions could be described as the following steps:

Step 1. Set $\lambda_{\min}$ and $\lambda_{\max}$. Initialize a population including m geese with random positions inside the solution space. Evaluate the fitness of each goose, and the one gets the best fitness is set as the leader whose index is set as l. The fitness of the leader is recorded as f (*leader*), and for the first time f (*leader*) is equal to the best fitness of the whole team in history which is recorded as f(*team*).

Step 2. Compute d_{il} ($1 \leqslant i \leqslant m$, $i \neq l$) for each goose in the team. If $d_{il} \leq 2\lambda_{min} l\sqrt{n}$, update the position of goose i as Eq. (3); if $d_{il} 2\lambda_{max} l\sqrt{n}$, update the position of goose i as Eq. (4); if $2\lambda_{min} l\sqrt{n} \leq d_{il} \leq 2\lambda_{max} l\sqrt{n}$, goose i, together with the leader, takes a linear search in the negative gradient direction of the leader.

Step 3. Evaluate the fitness of each goose and the one getting the best fitness is set as the leader whose index is set as l. Update f (*leader*).

Step 4. Compare f (*leader*) with f(*team*). If f (*leader*) is less than f(*team*), then the latter is updated with the former.

Step 5. If the stop criterion is met the algorithm ends; else go to Step 2.

The linear search in Step 2 above includes two basic steps: to determine the initial search interval and the step size. The golden section method is used to determine the step size. The initial search interval is determined by two different methods, the descending method and the hill striding method. The descending method is fit for unimodal functions. And both the two methods can be used in the process of optimizing multimodal functions. But it seems that the hill striding method has more chance to escape from local optimum regions on some functions, which we will discuss later. The two methods are described in detail here.

3.2.1.1. Descending method. The negative gradient direction of the leader may or may not be the descending direction of the followers. In the former case, we can find an interval containing a minimum point in the negative gradient direction of the leader; in the latter case we can also find an interval containing a minimum in the gradient direction of the leader. The steps could be described as follows:

1) Give a point x1 and the initial step size r, and let $x_2 = x_1 + r$. Compute $f(x_2)$. If $f(x_1) \geqslant f(x_2)$, go to step 2); else let $r = -r$.
2) Let r = 2r, $x_3 = x_2 + r$. Compute $f(x_3)$.
3) If $f(x_2) \geqslant f(x_3)$, let $x_1 = x_2$, $x_3 = x_2$, $f(x_1) = f(x_2)$, $f(x_2) = f(x_3)$, and then go to step (2); else, we get three points: $x1$, $x2$, $x3$, and $[x1, x3]$ is the initial linear search interval.

3.2.1.2. Hill striding method. When the negative gradient direction of the leader is not the descending direction of the followers, we do not try to find a descending direction by turning to the gradient direction of the leader. We attempt to stride over the maximum point (hill) in the negative direction of the leader and find an interval containing a minimum point behind the hill. The steps could be described as follows:

1) Give a point x_1 and the initial step size r, and let $x_2 = x_1 + r$. Compute $f(x_2)$. If $f(x_1) \geqslant f(x_2)$, go to step (2); else go to step (4).
2) Let r = 2r, $x_3 = x_2 + r$. Compute $f(x_3)$.
3) If $f(x_2) \geqslant f(x_3)$, let $x_1 = x_2$, $x_3 = x_2$, $f(x_1) = f(x_2)$, $f(x_2) = f(x_3)$, and then go to step (2); else we get three points: x_1, x_2, x_3, and $[x1, x3]$ is the linear search interval.
4) Let r = 2r, $x_3 = x_2 + r$, and compute $f(x_3)$.
5) If $f(x_2) < f(x_3)$, let $x_1 = x_2$, $x_3 = x_2$, $f(x_1) = f(x_2)$, $f(x_2) = f(x_3)$, and go to step (4); else, x_3 is just in a new descending interval (the hill has been stridden over).
6) Let r resumes its original value, and let $x_1 = x_3$, $f(x_1) = f(x_3)$, $x_2 = x_1 + r$. Compute $f(x_2)$. If $f(x_1) \geqslant f(x_2)$, go to step (2); else let $r = -r$, $x_2 = x_1 + r$ and compute $f(x_2)$. Then go to step (2).

4. Optimization experiment

4.1. Experimental setting

Four benchmarks [12,17,18], which are commonly used in the evolutionary computation, are selected as examples in this paper to compare goose team optimization with the standard particle swarm optimization. The formulations of the functions are listed below. And their numbers of dimensions (n), the admissible range of the variables, and the goal values are summarized in Table 1.

Formula of Sphere:

$$f_1(x) = \sum_{i=1}^{n} x_i^2$$

Formula of Rosenbrock:

$$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

Formula of Schaffer's f_6:

$$f_6(x) = 0.5 + \frac{\left(\sin\sqrt{x_1^2 + x_2^2}\right)^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$$

Formula of Griewank:

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \Phi_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}} + 1)$$

The Sphere function and the Rosenbrock function are two unimodal functions and have only one global minimum in the initial search space. The Rosenbrock function value changes very slightly in the long and narrow area close to the global minimum. The Schaffer's f_6 function and the Griewank function are two multimodal functions, which have very complicated landforms. The four functions are quite fit for the evaluation of algorithm performance.

We use two stop criterions in the experiment. Criterion 1 is to test whether the algorithm can find goal set for each function in 4000 iterations. Using the first criterion, we can get success rate, average iteration number, average run-

Table 1
Four benchmarks

Name	Dim n	Range$[x_{\min}, x_{\max}]^n$	Goal for f
Sphere	30	[−100, 100]	0.01
Rosenbrock	30	$[-30, 30]^n$	100
Schaffer's f_6	2	$[-100, 100]^2$	10^{-5}
Griewank	30	$[-100, 100]^n$	0.1

ning time and expected running time. Here expected running time is got as Eq. (5). Criterion 2 is to test whether the maximum iteration number is met. Using the second criterion we can get average optimum gained after 1000 iterations and average optimum gained after 4000 iterations.

$$\text{Expected running time} = \frac{\text{Average running time}}{\text{Success rate}} \tag{5}$$

The algorithms are programmed by java and run on a PC with P4-2.0GHz CPU and 256MB memory. The population size m is set to 10 and 30. The other parameters of GTO are set to $\lambda_{\min} = 0.01$, $\lambda_{\max} = 0.1$, and $r = 10^{-10}$. The other parameters of SPSO are set to $c_1 = c_2 = 2$, $\omega = 0.6$ with $m = 10$, and $\omega = 0.5$ with $m = 30$. Here, m represents the population size. The two algorithms have the best performance with the parameters obtained from large numbers of experiments. To get rid of the randomicity, the results are the average of 100 trial runs.

4.2. Results and discussion

The results of experiments of SPSO and GTO are listed in Table 2. Opt. 1 represents the average optimum gained after 1000 iterations. Opt. 2 represents the average optimum gained after 4000 iterations. Here, we define the global optimum zero as "less than10^{-45}". The average convergence curves of the two algorithms on the four functions with two population sizes over 100 trials are shown in Fig. 1. In the figure, the x coordinate is the iteration number; the y coordinate is $\lg f(x)$, the common logarithm of the fitness value, for the fitness value changes too much in the optimization process.

The Schaffer's f_6 function and the Griewank function are multimodal functions, and the hill striding method is used in the linear search. The Sphere function and the Rosenbrock function are unimodal functions and the descending method is used in the linear search. The results indicate that GTO has got a better performance than SPSO on the four benchmarks. Even with population size 10 the performance of GTO is satisfying, which implies that GTO has less population dependence. And this is the result of using gradient information of the function.

In GTO, we use two different methods to determine the initial search interval: the descending method and the hill striding method. It is obvious that we must use the descending method on a unimodal function. When optimizing a multimodal function, we can determine the initial search interval by either of the two methods. The descending method can find an interval containing an optimum faster than the hill striding method. Whereas the hill striding method has more chance to help the algorithm to escape from local optimum regions.

In the above experiments, the hill striding method is used in the linear search on the two multimodal functions: Schaffer's f_6 and Griewank. Next the descending method is used to perform some experiments on the two functions. The population size is set to 10 and 30. The other param-

Table 2
Performance of SPSO and GTO

m	Index	Algorithm	Sphere	Rosenbrock	Schaffer's f_6	Griewank
10	Success rate (%)	SPSO	1	0.73	0.75	0.95
		GTO	1	0.98	1	1
	Iterations	SPSO	804.0	1737.32	1808.33	1176.69
		GTO	1.04	164.79	422.84	1.01
	Ave. time (s)	SPSO	0.130	0.357	0.059	0.272
		GTO	0.009	0.038	0.021	0.009
	Exp. time (s)	SPSO	0.130	0.489	0.079	0.286
		GTO	0.009	0.039	0.021	0.009
	Opt. 1	SPSO	0.3581	2.952E4	6.154E-3	2.604
		GTO	0	32.52	4.066E-6	0
	Opt. 2	SPSO	5.940E-16	110.5	1.661	0.02677
		GTO	0	13.31	2.274E-7	0
30	Success rate (%)	SPSO	1	0.92	1	1
		GTO	1	1	1	1
	Iterations	SPSO	320.59	782.73	959.46	436.35
		GTO	1.06	111.31	43.61	1
	Ave. time (s)	SPSO	0.159	0.463	0.058	0.308
		GTO	0.010	0.038	0.010	0.010
	Exp. time (s)	SPSO	0.159	0.503	0.058	0.308
		GTO	0.010	0.038	0.010	0.010
	Opt. 1	SPSO	1.560E-11	188.5	2.122E-3	0.04225
		GTO	0	16.18	2.643E-8	0
	Opt. 2	SPSO	2.697E-28	52.97	1.912E-7	0.01393
		GTO	0	2.064	1.604E-9	0

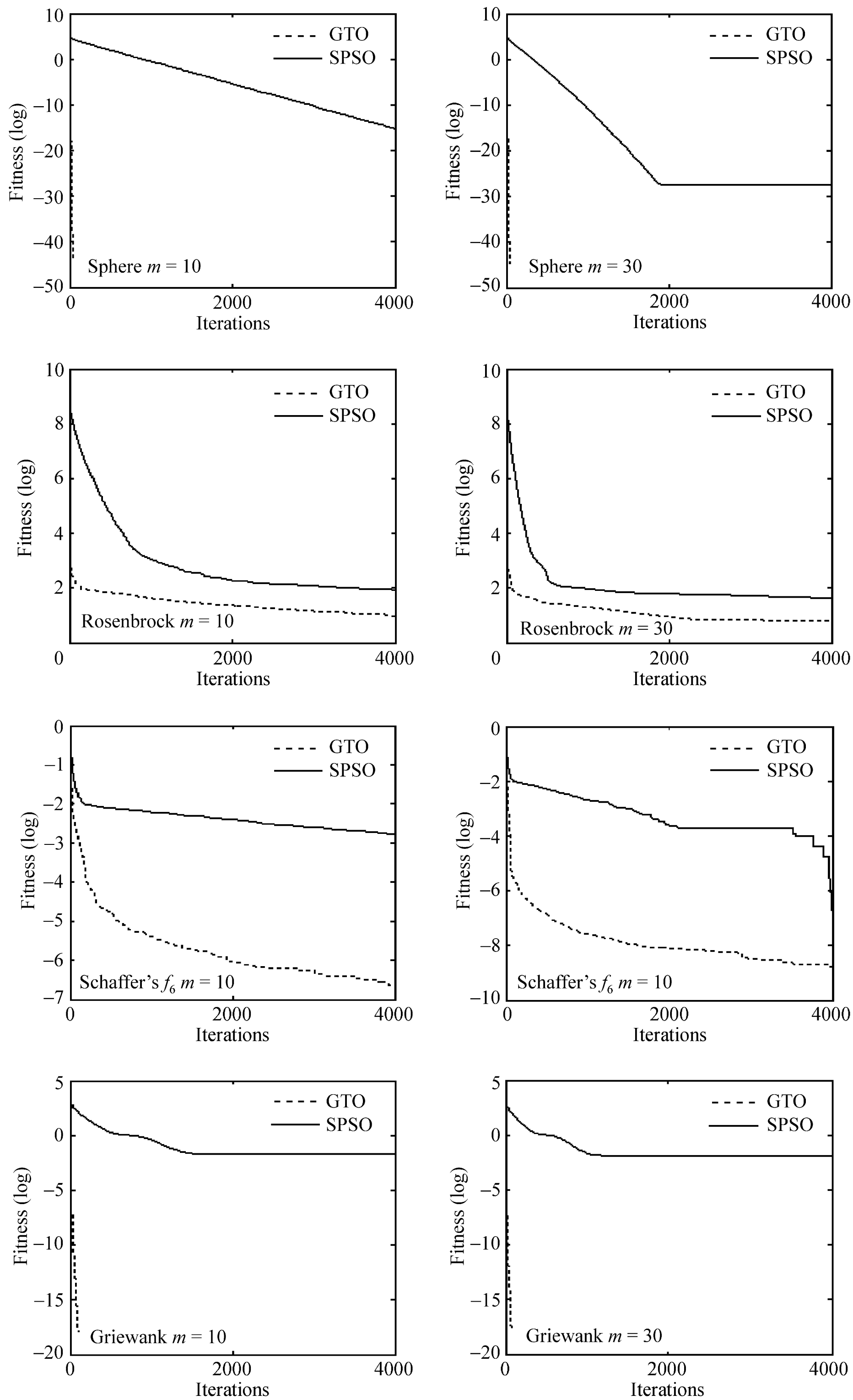

Fig. 1. The average convergence curves of SPSO and GTO.

eters are the same as above. To get rid of the randomicity the results are the average of 100 trial runs, which are listed in Table 3.

When comparing the results of Griewank in Tables 2 and 3, we can find that there is little difference between the performance of the two methods in the linear search. But the performance of GTO with the hill striding method on Schaffer's f6 is much better than that with the descending method. With the descending method, GTO can not reach the goal. The tremendous differences of the success rate between the two methods show that the hill striding method can make a goose team have more chance to escape

Table 3
Performance of GTO using descending method

m	Index	Schaffer's f_6	Griewank
10	Success rate (%)	0	1
	Iterations		1
	Ave. time (s)	0.100	0.011
	Exp. time (s)		0.011
	Opt. 1	0.1934	0
	Opt. 2	0.1934	0
30	Success rate (%)	0.03	1
	Iterations	153.0	1.2
	Ave. time (s)	0.177	0.009
	Exp. time (s)	5.9	0.009
	Opt. 1	0.03030	0
	Opt. 2	0.03030	0

from local optimum and help the goose team to reach the goal optimum faster.

In summary, both the descending method and the hill striding method can be used to determine the initial search interval on the multimodal functions. But sometimes the descending method will result in a bad performance. On the contrary, the hill striding method can provide the algorithm with more ability to escape from the local optimum regions. Of course, this advantage is not obvious for all the multimodal functions. It is problem-dependent.

5. Conclusion

There are different patterns in the aggregate motion of animals in the nature. And goose team is a special bird flock. By introducing the basic flight mechanism of a goose team, such as division of roles, separate principle, parallel principle and aggregate principle, into the simple social model of the standard PSO, the particle swarm optimization with leader and followers, namely goose team optimization, is constructed. Goose team optimization is an attempt to add different animal aggregation behaviors to optimization technique. The goose team flight mechanism enables the algorithm to optimize four complicated nonlinear functions very effectively. Using the gradient information is helpful to consider the inherent feature of the problem and to decrease the population dependence of the algorithm. In general, GTO of 10 individuals can get satisfying performance.

Future research will be done on goose team optimization algorithm. The use of gradient information restricts the application scope to a certain extent. The authors wish to find other ways for the goose leader to determine the flying direction. In addition more test and analysis of different problems, especially real-world applications, need to be done on GTO.

Acknowledgements

The work was supported by National Natural Science Foundation of China (Grant Nos. 70431003 and 60521003) and National Key Technology R&D Program (Grant No. 2006BAH02A09).

References

[1] Holland JH. Adaptation in nature and artificial systems. Ann Arbor: University of Michigan; 1975.
[2] Farmer JD, Packed NH. The immune system, adaptation, and machine learning. Physica 1986;22D(2):187–204.
[3] Hopfield JJ, Tank DW. Neural computation of decision in optimization problems. Biol Cybern 1985;52:141–52.
[4] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 1996;26(1):29–41.
[5] Reynolds RG, Rolnick SR. Learning the parameters for a gradient-based approach to image segmentation from the results of a region growing approach using cultural algorithms. In: Proceedings of international conference on evolutionary computation. USA: IEEE Service Center; 1995. p. 819–24.
[6] Wang DW. Colony location algorithm for assignment problems. J Control Theory Appl 2004;2(2):111–6.
[7] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Australia: Perth; 1995. p. 1942–8.
[8] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings on international symposium on micro machine and human science. USA: IEEE Service Center; 1995. p. 39–43.
[9] Kennedy J. The particle swarm: social adaptation of knowledge. In: Proceedings of IEEE international conference on evolutionary computation. USA: IEEE Service Center; 1997. p. 303–8.
[10] Millionas MM. Swarm, phase, transition, and collective intelligence. In: Artificial Life. MA: Addison Wesley; 1994.
[11] Suganthan PN. Particle swarm optimiser with neighborhood operator. In: Proceedings of the congress on evolutionary computation. USA: IEEE Service Center; 1999. p. 1958–62.
[12] Trelea IC. The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 2003;85: 317–25.
[13] Reynolds CW. Flock, herds and schools: a distributed behavioral model. Comput Graph 1987;21(4):25–34.
[14] Shaw E. Schooling in fishes: critique and review. In: Development and evolution of behavior. W.H. San Francisco: Freeman and Company; 1970. p. 452–80.
[15] Anderson C, Franks NR. Teams in animal societies. Behav Ecol 2001;12(5):534–40.
[16] Lissaman PBS, Shollenberger CA. Formation flight of birds. Science 1970;168(3934):1003–5.
[17] Clerc M, Kennedy J. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space. IEEE Trans Evol Comput 2002;6(1):58–73.
[18] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: Proceedings on IEEE international conference on evolutionary computation. USA: IEEE Service Center; 1998. p. 69–73.