Nonlinear Dyn (2012) 68:129–136
DOI 10.1007/s11071-011-0209-x

ORIGINAL PAPER

Chaotic ant swarm optimization with passive congregation

Yuying Li · Qiaoyan Wen · Baihong Zhang

Received: 15 March 2011 / Accepted: 23 August 2011 / Published online: 22 September 2011

Abstract Chaotic ant swarm optimization (CASO) is a powerful chaos search algorithm for optimization problems, but it is often easy to be premature convergence. To overcome the weakness, this paper presents a CASO with passive congregation (CASOPC). Passive congregation is one type of biological information sharing mechanisms that allow animals to aggregate into groups and help to enhance the exploitation of animals. By introducing passive congregation strategy into the CASO, a modified evolution equation based on the CASO is proposed in the CASOPC. The modified evolution equation cannot only employ the parallel search of all ants and the well exploration ability of the CASO, but also stress and control the exploitation by passive congregation coefficient c in the stage of evolution. Due to linearly increasing c in the CASOPC, the exploration and exploitation ability of ants are well balanced so that premature convergence can be avoided and good performance can be achieved. In order to estimate the capability of the CASOPC, it is tested with a set of 5 benchmark functions with 30 dimensions and compared to the CASO. Experimental results indicate that the CASOPC improves the search performance on the benchmark functions significantly.

Keywords Chaos search · Optimization problems · Chaotic ant swarm optimization · Passive congregation

Y. Li (✉) · B. Zhang
Institute of Chemical Defense of the Chinese People's Liberation Army, Beijing 102205, China
e-mail: yuying_20042009@yahoo.com.cn

Q. Wen
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

1 Introduction

In the past few decades, nature-inspired computation has received increasing attention and wide applications in a variety of fields [1]. Nature serves as a fertile source of concepts, principles, and mechanisms for designing artificial intelligence computation to handle complex computational problems. Nature-inspired computation includes mainly evolutionary algorithms (EAs) [2] and swarm intelligence (SI) [3–5]. The first one draws inspiration from evolution by natural selection. Another one is inspired by collective animal behavior. Since EAs and SI were introduced, they have been applied to many optimization problems successfully and many modifications have been proposed. Especially, due to simple mechanism and high performance for global optimization, a popular SI paradigm, namely chaotic ant swarm optimization (CASO) [6], gradually becomes a hot topic. Until now, the CASO has been applied in many practical systems [7–9].

The CASO, which is inspired by foraging actions of ant colony, is a chaos search algorithm to tackle optimization problems. In the CASO, the ants use chaotic principles to search for food. Each ant performs a chaotic exploration of its hunting sites and interacts with its neighbors. They search chaotically until they have been organized via pheromone trails, and move to the site which is the most successful among the previously met hunting sites. The principles can be used to implement a meta-heuristic for the search of a global optimum or near-optimum of a function in a search space where each ant represents a solution to the problem being optimized. Therefore, the CASO can find optimal regions of complex search spaces through the interaction of individuals in ant swarm.

Similar to other nature-inspired computation, such as EAs, the CASO is often easy to be premature convergence. This requires us to consider the development of a more effective CASO-based algorithm rather than simple application of pure CASO. As for the CASO, some modifications have been proposed recently [10]. In 1997, biologists proposed four types of biological mechanisms that allow animals to aggregate into groups: passive aggregation, active aggregation, passive congregation, and social congregation [11]. There are different information sharing mechanisms inside these forces. Passive congregation has been introduced to particle swarm optimization (PSO)[12] and improves the search performance of the standard PSO. The research [12] about PSO with passive congregation indicates information can be transferred among individuals that will help individuals to avoid misjudging information and becoming trapped by poor local minima by introducing passive congregation. Moreover, we found the passive congregation is suitable to be incorporated in the CASO. To improve the search ability of the CASO, this paper proposes a hybrid algorithm of the CASO with passive congregation to achieve results with high quality and reliance.

The rest of the paper is organized as follows. Section 2 introduces the CASO. A CASO with passive congregation is presented in Sect. 3. In Sect. 4, we describe the test functions, experimental settings, and the experimental results. The conclusion is given in Sect. 5.

2 Chaotic ant swarm optimization

Since many search strategies based on chaos have been found to obtain nice capabilities of hill-climbing and escaping from local optima, and to be more effective than random search [13], chaotic dynamics has received particular attention. In 1991, Cole pointed out that ant colony exhibits a periodic behavior while single ant shows low-dimensional deterministic chaotic activity patterns [14]. From the view of dynamics, the chaotic behavior of single ant has some relation to the self-organizing and foraging behaviors of ant colony. The chaotic behavior of individual ant and the intelligent organization actions of ant colony are adaptations to the environment. These behaviors help the ants to find food and survive. According to the theory of chaotic dynamics and optimization principles, a novel optimization algorithm, called chaotic ant swarm optimization (CASO), was presented.

In the CASO, the chaotic system $x(k+1) = x(k) \times e^{\mu(1-x(k))}$ [15] was introduced into the heuristic equation of the CASO for obtaining the chaotic search initially. The adjustment of the chaotic behavior of individual ant is achieved by the introduction of a successively decrement of organization variable y_i and leads the individual to move to the new site acquired with the best fitness value eventually. $(\text{pbest}_{\text{id}} - x_{\text{id}})$ is introduced to achieve the information exchange of individuals and the movements to new site taken on the best fitness value. pbest_{id} is selected based on the fitness theory which is very widely developed in optimization theory such as genetic algorithm and tabu search, and so on. x_{id} is the location of the dth dimension of ant i.

The CASO is a kind of iterative optimization algorithm, which is firstly employed in the optimization of sequential space. In the sequential space coordinates, the mathematical description [6] of the CASO as follows:

$$\begin{cases} y_i(k+1) = y_i(k)^{(1+r_i)} \\ x_{\text{id}}(k+1) \\ \quad = \left(x_{\text{id}}(k) + \frac{7.5}{\psi_d} \times v_i\right) \\ \qquad \times e^{(1-e^{-ay_i(k+1)})(3-\psi_d(x_{\text{id}}(k)+\frac{7.5}{\psi_d}\times v_i))} \\ \qquad - \dfrac{7.5}{\psi_d} \times v_i \\ \qquad + \left(\text{pbest}_{\text{id}}(k) - x_{\text{id}}(k)\right)e^{-2ay_i(k+1)+b} \end{cases} \tag{1}$$

where the superscripts k and $k+1$ denote the time index of the current and the next iteration, respectively; $y_i(k)$ is the ith ant's organization variable of the current iteration step, $y_i(1) = 0.999$; $\text{pbest}_{\text{id}}(k)$ is the best location found by the ith ant and its neighbors within k steps; v_i $(0 < v_i < 1)$ determines the search region of ant i; a is a sufficiently large positive constant and can be selected as $a = 200$; $b(0 \le b \le 2/3)$ is a constant; $x_{\mathrm{id}}(k)$ is the current location of the dth dimension of ant i, $x_{\mathrm{id}}(1) = \frac{7.5}{\psi_d} \times (1 - v_i) \times \mathrm{rand}(1)$, where rand(1) is a uniformly distributed random number in (0, 1).

r_i and ψ_d are two important parameters. r_i is the organization factor of ant i, which affects the convergence speed of the CASO directly. If r_i is very large, the iteration step of "chaotic" search is small then the system converges quickly and the desired optima or near-optima cannot be achieved. If r_i is very small, the iteration step of "chaotic" search is large then the system converges slowly and the runtime will be longer. Since small changes are desired as iteration step evolves, the value of r_i is chosen typically as $0 < r_i \le 0.5$. The format of r_i can be designed according to concrete problems and runtime. Each ant could have different r_i, such as $r_i = 0.2 + 0.03 \times \mathrm{rand}(1)$. ψ_d affects the search ranges of the CASO. If the interval of the search is $[-\frac{\omega_d}{2}, \frac{\omega_d}{2}]$, we can obtain an approximate formula $\omega_d \approx \frac{7.5}{\psi_d}$.

In the CASO, the neighbors of the ant are defined finite ants according to their distance in space. Due to the influence of self-organization behaviors of ants, the impact of organization will become stronger than before and the neighbors of the ant will increase. That is to say, the number of nearest neighbors is dynamically changed as iterative steps increase. In this paper, we use this kind of dynamical neighbors in which the number q of single ant increases for every T iterative steps. In order to simulate the behaviors of ants, we use the Euclidean distance. Supposing there are two ants whose locations are $(x_{i1}, \ldots, x_{iD})$ and $(x_{j1}, \ldots, x_{jD})$ respectively, where $i, j = 1, \ldots, L$ (here, L is the size of ant swarm) and $i \ne j$, the distance between the two ants is $\sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{iD} - x_{jD})^2}$.

Equation (1) describes the search process of the CASO. The organization variable y_i is used to control the chaotic process of ant moving, and its influence on the ant's behavior is very weak initially. That is, initially the organization capabilities of the ants are very weak so that a noncoordinated process occurs which is characterized by the chaotic walking of ants. This phase lasts until the influence of organization on the individual behavior is sufficiently large. Then the chaotic behavior of the individual ant disappears and a coordination phase starts. That is, ants do some further searches and move to the best location which they have ever found in search space. Throughout the whole process, these ants exchange information with other ants, then compare and memorize the information.

The procedure of the CASO can be summarized as follows.

Step 1: Randomly initialize the locations and organization factors of all ants.

Step 2: Evaluate fitness values of all ants, let the pbest of each ant and its fitness value equal to its current location and fitness value.

Step 3: Update organization variable and location vector according to (1) for each ant.

Step 4: Evaluate fitness values of all ants, and select the pbest of each ant.

Step 5: If a predefined stopping criterion is met, then select and output the best ant with minimal fitness value in ant swarm and its fitness value; otherwise go back to Step 3.

3 Chaotic ant swarm optimization with passive congregation

As stated before, the CASO has greater exploration ability, but the exploitation ability around the optimum is not very good. How can we enhance the exploitation ability of the CASO? Biologists have indicated that passive congregation is an attraction of an individual to other group members but where there is no display of social behavior [11]. That is, it is one type of biological information sharing mechanisms that allow animals to aggregate into groups and help to enhance the exploitation of animals. Thus, passive congregation is suitable to be incorporated in the CASO model, which is called chaotic ant swarm optimization with passive congregation (CASOPC). In the CASOPC, the ant swarm is manipulated according to the following equations:

$$\begin{cases} y_i(k+1) = y_i(k)^{(1+r_i)} \\ x_{\mathrm{id}}(k+1) = \left(x_{\mathrm{id}}(k) + \dfrac{7.5}{\psi_d} \times v_i\right) \\ \qquad \times e^{(1-e^{-a y_i(k+1)})(3-\psi_d(x_{\mathrm{id}}(k)+\frac{7.5}{\psi_d}\times v_i))} \\ \qquad - \dfrac{7.5}{\psi_d} \times v_i \\ \qquad + c \times \left(R_{\mathrm{id}}(k) - x_{\mathrm{id}}(k)\right) \\ \qquad \times e^{-2a y_i(k+1)+b} \\ \qquad + (1-c) \times \left(\mathrm{pbest}_{\mathrm{id}}(k) - x_{\mathrm{id}}(k)\right) \\ \qquad \times e^{-2a y_i(k+1)+b} \end{cases} \tag{2}$$

where $R_{\mathrm{id}}(k)$ is an ant randomly selected from the swarm at the current iteration step. c is the passive congregation coefficient which is used to adjust the exploration and exploitation of ants. It can be a positive constant or even a positive linear or nonlinear function of time. This value of c is discussed in Sect. 4.2. In order to reduce the likelihood of the ant leaving the search space, the value of $x_{\mathrm{id}}(k+1)$ is usually chosen as $x_{\mathrm{id}}(k+1) = \min(\frac{\omega_d}{2}, \max(-\frac{\omega_d}{2}, x_{\mathrm{id}}(k+1)))$.

The pseudocode for the CASOPC is described as follows.

Step 1: Randomly initialize the locations and organization factors of all ants.
Step 2: Evaluate fitness values of all ants, let the *pbest* of each ant and its fitness value equal to its current location and fitness value.
Step 3: Randomly choose an ant as R_i.
Step 4: Update organization variable and location vector according to (2) for each ant, and restrict $x_{\mathrm{id}}(k+1) = \min(\frac{\omega_d}{2}, \max(-\frac{\omega_d}{2}, x_{\mathrm{id}}(k+1)))$.
Step 5: Evaluate fitness values of all ants, and select the pbest of each ant.
Step 6: If a predefined stopping criterion is met, find the ant with minimal fitness value in all ants, noted as gbest; otherwise go back to Step 3.
Step 7: Perform discrete recombination operator (see Fig. 1), then output gbest that is the best ant location with minimal fitness.

It can be seen from the above procedure that the CASOPC cannot only retain the advantage of the CASO, but also have three differences from the CASO. The advantage is that all ants are analyzed in parallel at each step from step 2. Three differences are as follows. Firstly, from (2), instead of learning from the pbest ant, an ant cannot only learn from the *pbest* ant, but also an random ant in order to adjust the exploration and exploitation of ants. Secondly, in step 4, $x_{\mathrm{id}}(k+1) = \min(\frac{\omega_d}{2}, \max(-\frac{\omega_d}{2}, x_{\mathrm{id}}(k+1)))$ is applied to prevent ants moving out of the search bounds after using (2). Finally, from step 6, the discrete recombination operator is employed to improve solution quality further after stopping criterion is met.

In addition, the above CASOPC can be transferred to two simpler approaches. For example, if $c = 0$ in (2), each ant can only learn from the corresponding *pbest* to make the ant generate new location, obviously, the CASOPC is changed into the CASO described in Sect. 2. On the other hand, if $c = 1$ in (2), each ant can only learn form a random ant, that is, the CASOPC makes the ant swarm fail to search for a good location.

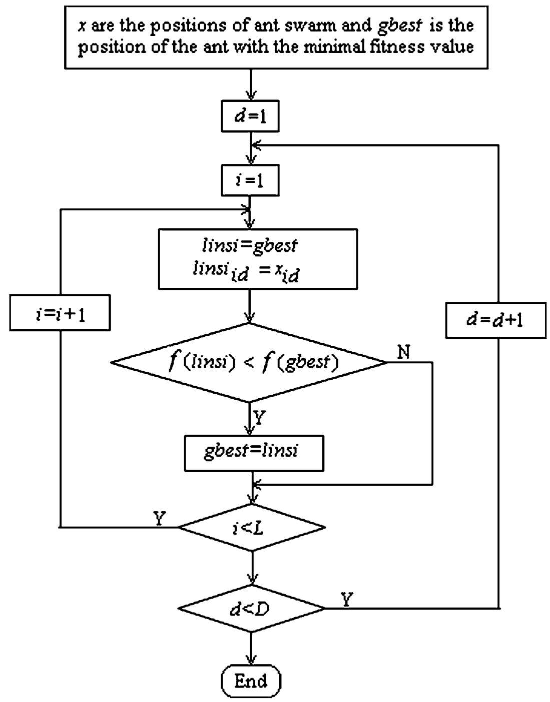

Fig. 1 Flowchart of the discrete recombination operator

4 Experimental studies

In the CASOPC, the newly introduced passive congregation coefficient c is important for balancing the exploration and exploitation of ant swarm. Thus, the value of c should be suitably selected according to the concrete optimization problems. The section discusses the influence of c on the search result of the CASOPC, and compares the CASOPC with the CASO.

4.1 Test functions

In our experimental studies, a set of 5 benchmark functions, with the global minimum fitness value 0 (this paper defined $f_i(x)$ $(i = 1, 2, 3, 4, 5)$ as fitness function), was employed to evaluate the CASOPC in comparison with the CASO.

The first function is the Sphere function:

$$f_1(x) = \sum_{i=1}^{D} x_i^2, \quad \text{s.t.} \quad x_i \in [-50, 50]. \tag{3}$$

The second function is the DeJongF4 function:

$$f_2(x) = \sum_{i=1}^{D} i x_i^2, \quad \text{s.t.} \quad x_i \in [-20, 20]. \tag{4}$$

The third test function is the Rosenbrock function:

$$f_3(x) = \sum_{i=1}^{D-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right),$$
$$\text{s.t.} \quad x_i \in [-100, 100]. \tag{5}$$

The fourth test function is the Griewank function:

$$f_4(x) = 1 + \sum_{i=1}^{D} \left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt[2]{i}}\right),$$
$$\text{s.t.} \quad x_i \in [-600, 600]. \tag{6}$$

The fifth test function is the Rastrigin function:

$$f_5(x) = \sum_{i=1}^{D} \left(10 + x_i^2 - 10\cos(2\pi x_i)\right),$$
$$\text{s.t.} \quad x_i \in [-5.12, 5.12]. \tag{7}$$

The above benchmark functions were tested widely by EAs and SI, such as evolutionary programming, simulated annealing, genetic algorithms, and particle swarm optimization. The first two are unimodal while the last two are multimodal. Rosenbrock function is multimodal as soon as the dimension is more than 3. Sphere function is an easy, unimodal function that any optimization technique should be able to solve with a good degree of resolution. It helps to identify good local optimizers. DeJongF4 function is unimodal. Rosenbrock function is generally difficult to optimize even for gradient-based algorithms. The last two are multimodal functions where the number of local minima increases exponentially with the problem dimension. These two functions help to test the global optimization capabilities of tested algorithms.

4.2 Experimental setting

To evaluate the performance of the proposed CASOPC, the CASO [6] were used for comparisons. The parameters to the two algorithms ((1) and (2)) are set: $a = 200$, $b = \frac{1}{2}$, $y(1) = 0.999$, $r_i = 0.01 + 0.00001 \times \text{rand}(1)$, $v_i = \text{rand}(1)$, $D = 30$, $L = 20$. This kind of dynamical neighbor is selected. At the first step, the number of neighbors is two. The number of neighbors will increase one every two iterative steps. The max number of neighbors is 19. The value of parameter ψ_d can be selected according to the ranges of intervals [6]. A fixed number of maximum iterations 1000 was applied. All experiments were repeated for 50 runs and the results were averaged to account for stochastic differences. The experiments were performed on a computer with 2.93 GHz Intel(R) Pentium(R) 4 processor and 512 MB of RAM using Matlab 7.6.

Because the newly introduced passive congregation coefficient c is crucial for enhancing the search performance of the CASO, experiments were executed to select a proper value of c. All functions were tested with different values of c. The average test results obtained from 50 runs are listed in Table 1 and 3.81E−01 is defined 3.81×10^{-1}. From Table 1, when $c = 0.6$, the CASOPC generated good results on Sphere function and Rosenbrock function. When $c = 0.5$, on DeJongF4 function. For Griewank function, the best results were generated at the point $c = 0.4$. For the Rastrigin function, at the point $c = 0.7$. When $c = 1.0$, the search performance of the CASOPC on all functions is deteriorated from Table 1. Therefore, a generic c for all functions should be equal or smaller than 0.7.

It is our interest to investigate whether the CASOPC with a linear increasing c generates better results on the benchmark functions than the CASOPC with a fixed value of c. Therefore, Rastrigin function was selected and tested with different ranges of linearly increasing c. The results are tabulated in Table 2. The best result was generated by the CASOPC with a linearly increasing passive congregation coefficient c which started at 0.0 and ended at 0.6, and it was better than the best result generated by the CASOPC with different fixed value of c. From Table 1 and Table 3, we could see that linearly increasing c from 0.0 to 0.6 gives the CASOPC the better performance compared with all fixed c values on most of five benchmark functions. The exception is Sphere function because the best mean value is achieved when $c = 0.6$.

Table 1 Average fitness values of all test functions with different c

Value of c	Sphere	DeJongF4	Rosenbrock	Griewank	Rastrigin
$c = 0.0$	3.81E−01	1.61E−02	2.34E+01	4.66E−01	2.26E+01
$c = 0.1$	5.20E−03	1.83E−05	3.13E+01	1.12E−01	2.48E−02
$c = 0.2$	5.20E−03	1.83E−05	3.77E+01	1.58E−01	1.34E−02
$c = 0.3$	9.38E−03	1.34E−05	4.26E+01	1.56E−01	2.61E−02
$c = 0.4$	6.93E−03	1.17E−05	2.78E+01	6.69E−02	9.80E−03
$c = 0.5$	3.49E−03	4.79E−06	2.89E+01	8.18E−02	1.58E−02
$c = 0.6$	3.44E−03	8.51E−04	2.48E+01	1.00E−01	7.23E−03
$c = 0.7$	1.27E−02	8.55E−06	3.59E+01	1.04E−01	6.28E−03
$c = 0.8$	3.47E−03	1.85E−05	8.82E+01	1.34E−01	1.00E−02
$c = 0.9$	3.96E−02	1.84E−04	3.24E+02	2.54E−01	9.90E−02
$c = 1.0$	1.95E+04	1.11E+07	8.08E+10	8.37E+02	2.27E+02

Table 2 Average fitness value of Rastrigin function with different linearly increasing c

c_{max}	c_{min}							
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0.1	7.44E−03	NA	NA	NA	NA	NA	NA	NA
0.2	1.11E−02	9.06E−03	NA	NA	NA	NA	NA	NA
0.3	1.87E−01	1.41E−02	4.43E−02	NA	NA	NA	NA	NA
0.4	1.02E−02	1.95E−02	1.88E−02	1.23E−02	NA	NA	NA	NA
0.5	9.56E−03	9.37E−03	8.02E−03	9.18E−03	1.00E−02	NA	NA	NA
0.6	3.49E−03	1.05E−02	7.26E−03	2.13E−02	1.21E−02	1.30E−02	NA	NA
0.7	2.26E−02	8.33E−03	7.84E−03	8.32E−03	1.79E−02	8.85E−03	9.96E−03	NA
0.8	8.08E−03	1.34E−02	1.12E−02	2.58E−02	1.74E−01	1.88E−02	1.93E−01	3.21E−02
0.9	8.80E−03	3.29E−02	3.32E−02	6.67E−03	3.18E−02	6.43E−03	3.27E−02	7.03E−02

Table 3 Comparison between the CASOPC and the CASO

Function	Mean		Variance		Mean runtime (s)	
	CASOPC	CASO	CASOPC	CASO	CASOPC	CASO
Sphere	6.97E−03	3.81E−01	3.35E−04	5.33E−02	81.24	80.15
DeJongF4	3.34E−06	1.61E−02	6.19E−11	1.62E−03	88.39	87.30
Rosenbrock	2.17E+01	2.34E+01	7.70E+02	1.37E+04	84.72	83.26
Griewank	5.44E−02	4.66E−01	2.27E−02	1.82E−01	90.89	89.78
Rastrigin	3.49E−03	2.26E+01	7.25E−05	1,10E+03	86.76	85.45

Although linearly increasing c gets worse mean value than $c = 0.6$ for Sphere function, still it achieves the result in the same rank as $c = 0.6$. Moreover, linearly increasing c tends to have more global search ability at the beginning of the run while having more local search ability near the end of the run. Thus, we choose linearly increasing c from 0.0 to 0.6 in this paper.

4.3 Experimental results and comparison

The experimental results (i.e., the mean and the deviation of the fitness values found in 50 runs) for the CASOPC and the CASO on each test function are listed in Table 3. All the settings are the same as mentioned in Sect. 4.2. Here, the results of the CASO are from [6].

From Table 3, we observe that the CASOPC achieves better results in the mean and variance than the CASO on all five test functions because of employing passive congregation strategy in the CASOPC. Especially, four orders of magnitude are improved in the mean while eight orders of magnitude in the variance for DeJongF4 and Rastrigin function. Although Rosenbrock function has not change the order of magnitude in the mean, it improves two orders of magnitude in the variance. That is, the stability of Rosenbrock function is enhanced in the CASOPC. For others functions, the mean and variance both are improved. Such as, Sphere function both increases two orders of magnitude in the mean and variance. Griewank function both improves one order of magnitude in the mean and variance. According to the mean, the CASOPC is more efficient to improve solution search ability than the CASO. Such as, from the results of Sphere function, the CASOPC increases the local search ability of the CASO; as seen in Griewank and Rastrigin function, the CASOPC enhances the global search ability of the CASO. Depending on the variance, solution stability is enhanced in the CASOPC compared with the CASO. Thus, the CASOPC improves the performance of the CASO in solution search ability and solution stability.

From Table 3, note that the CASOPC takes a little higher mean runtime than the CASO because of using passive congregation strategy in the CASOPC. However, from experimental results (see Table 3), we could see that the CASOPC outperforms the CASO on all five benchmark functions. So, we could say it is efficient. Moreover, with the rapid development of computer, the tradeoff between high-quality solutions and computational time tends to the former. So, the quality of solutions preponderates when problems could be solved by algorithms in rational time.

In order to more clearly observe the convergence trend of the CASOPC and the CASO, the two algorithms are run for 1500 iterations in Figs. 2, 3, 4, 5, 6. These figures show the convergence curve of the CASOPC and the CASO in one run for each test function. And we can see that the CASOPC has better search ability and solution stability than the CASO from these figures.

In a word, the overall results of Table 3 and Figs. 2–6 substantiate our claim that the introduction of passive congregation strategy enhances the performance of the CASO.

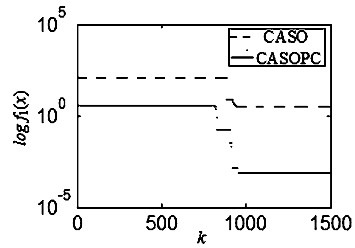

Fig. 2 Convergence curves for Sphere function

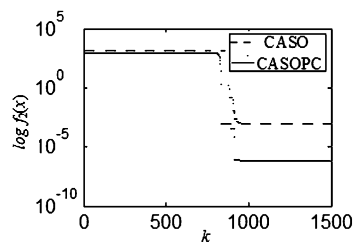

Fig. 3 Convergence curves for DeJongF4 function

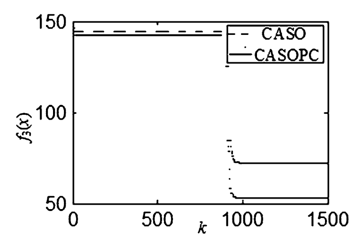

Fig. 4 Convergence curves for Rosenbrock function

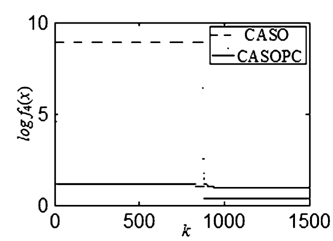

Fig. 5 Convergence curves for Griewank function

5 Conclusion

By simply incorporating passive congregation strategy into the CASO, we propose an effective hybrid algorithm named CASOPC for optimization problems.

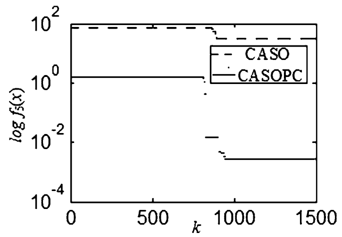

Fig. 6 Convergence curves for Rastrigin function

On one hand, the parallel searching structure and the well exploration ability of the CASO are employed; on the other hand, the exploitation is stressed in the stage of evolution and is controlled by passive congregation coefficient c. Due to linearly increasing of c, exploration and exploitation ability are well balanced so that premature convergence can be avoided and good performance can be achieved. Simulation results and comparisons based on five benchmark functions demonstrate the effectiveness and efficiency of the CASOPC for optimization problems. The future work is to apply the CASOPC for other kinds of optimization problems including real applications.

Acknowledgements The authors would like to thank the anonymous referees for their careful reading of the manuscript and their comments and suggestions. This work is supported by National Natural Science Foundation of China (Grant Nos. 60873191, 60903152, 61003286, 60821001), the Fundamental Research Funds for the Central Universities (Grant Nos. BUPT2011YB01, BUPT-2011RC0505).

References

1. Fogel, D.B.: The advantages of evolutionary computation. In: Proc. BCEC'97, Sweden vol. 1, pp. 1–11. World Scientific, Singapore (1997)
2. Barnard, C.J., Sibly, R.M.: Producers and scroungers: a general model and its application to captive flocks of house sparrows. Anim. Behav. **29**, 543–550 (1981)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems, 1st edn. Oxford University Press, London (1999)
4. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence, 1st edn. San Mateo, Morgan Kaufmann (2001)
5. Engelbrecht, A.: Fundamentals of Computational Swarm Intelligence, 1st edn. Wiley, New York (2005)
6. Li, L., Yang, Y., Peng, H., Wang, X.: An optimization method inspired by chaotic ant behavior. Int. J. Bifurc. Chaos **16**, 2351–2364 (2006)
7. Cai, J., Ma, X., Li, L., Yang, Y., Peng, H., Wang, X.: Chaotic ant swarm optimization to economic dispatch. Electr. Power Syst. Res. **77**, 1373–1380 (2007)
8. Li, L., Yang, Y., Peng, H.: Fuzzy system identification via chaotic ant swarm. Chaos Solitons Fractals **40**, 1399–1407 (2009)
9. Zhu, H., Li, L., Zhao, Y., Guo, Y., Yang, Y.: CAS algorithm-based optimum design of PID controller in AVR system. Chaos Solitons Fractals **42**, 792–800 (2009)
10. Li, Y., Wen, Q., Li, L., Peng, H.: Hybrid chaotic ant swarm optimization. Chaos Solitons Fractals **42**, 880–889 (2009)
11. Parrish, J.K., Hamner, W.M.: Animal Groups in Three Dimensions. Cambridge University Press, Cambridge (1997)
12. He, S., Wu, Q.H., Wen, J.Y., Saunders, J.R., Paton, R.C.: A particle swarm optimizer with passive congregation. Biosystems **78**, 135–147 (2004)
13. Hayakawa, Y., Marumoto, A., Sawada, Y.: Effects of the chaotic noise on the performance of a neural network model for optimization problems. Phys. Rev. E **51**, 2693–2696 (1995)
14. Cole, B.J.: Is animal behavior chaotic? Evidence from the activity of ants. Proc. R. Soc. Lond. B, Biol. Sci. **244**, 253–259 (1991)
15. Solé, R.V., Miramontes, O., Goodwill, B.C.: Oscillations and chaos in ant societies. J. Theor. Biol. **161**, 343–357 (1993)