Applied Mathematics and Computation 217 (2010) 3166–3173

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Gbest-guided artificial bee colony algorithm for numerical function optimization

Guopu Zhu [a], Sam Kwong [b,*]

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, PR China
[b] Department of Computer Science, City University of Hong Kong, Hong Kong SAR, PR China

ARTICLE INFO

Keywords:
Artificial bee colony algorithm
Genetic algorithm
Particle swarm optimization
Differential evolution
Biological-inspired optimization algorithm
Numerical function optimization

ABSTRACT

Artificial bee colony (ABC) algorithm invented recently by Karaboga is a biological-inspired optimization algorithm, which has been shown to be competitive with some conventional biological-inspired algorithms, such as genetic algorithm (GA), differential evolution (DE) and particle swarm optimization (PSO). However, there is still an insufficiency in ABC algorithm regarding its solution search equation, which is good at exploration but poor at exploitation. Inspired by PSO, we propose an improved ABC algorithm called gbest-guided ABC (GABC) algorithm by incorporating the information of global best (gbest) solution into the solution search equation to improve the exploitation. The experimental results tested on a set of numerical benchmark functions show that GABC algorithm can outperform ABC algorithm in most of the experiments.

1. Introduction

By now, there have been several kinds of biological-inspired optimization algorithms, such as genetic algorithm (GA) inspired by the Darwinian law of survival of the fittest [1,2], particle swarm optimization (PSO) inspired by the social behavior of bird flocking or fish schooling [3,4], ant colony optimization (ACO) inspired by the foraging behavior of ant colonies [5], and Biogeography-Based Optimization (BBO) inspired by the migration behavior of island species [6]. By simulating the foraging behavior of honey bee swarm, Karaboga [7] recently invented a new kind of optimization algorithm called artificial bee colony (ABC) algorithm for numerical function optimization. A set of experimental results on function optimization [8–11] show that ABC algorithm is competitive with some conventional biological-inspired optimization algorithms, such as GA, differential evolution (DE) [12], and PSO.

Since its invention in 2005, ABC algorithm has been applied to solve many kinds of problems besides numerical function optimization. In [13], Singh applied ABC algorithm for the Leaf-Constrained Minimum Spanning Tree (LCMST) problem. The experimental results presented in [13] show that comparing with GA, ACO and Tabu Search (TS), ABC algorithm can obtain better quality solutions of the LCMST problem in shorter time. Karaboga [14] used ABC algorithm to design Infinite Impulse Response (IIR) filters. And the performance of ABC algorithm was compared with that of a conventional optimization algorithm (LSQ-nonlin) [15] and PSO algorithm in the designs of IIR filters. According to their experimental results, ABC algorithm can be an alternative to design low-and high-order digital IIR filters [14]. Rao et al. [16] also applied ABC algorithm to solve the distribution system loss minimization problem. Their simulation results on the optimization of distribution network configuration show that ABC algorithm outperforms GA, DE and simulated annealing in terms of the quality of solution and the computation efficiency. Furthermore, ABC algorithm was also applied in the training of neural networks [17], the

* Corresponding author.
E-mail address: cssamk@cityu.edu.hk (S. Kwong).

0096-3003/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2010.08.049

parameter optimization of milling process [18], the optimization of constrained problems [19], the lot-streaming flow shop scheduling problem [20], and so on.

According to the various applications mentioned above, ABC algorithm seems to be a well-performed optimization algorithm. However, there is still an insufficiency in ABC algorithm regarding the solution search equation, which is used to generate new candidate solutions of ABC algorithm based on the information of previous solutions. It is well known that both exploration and exploitation are necessary for a population-based optimization algorithm. In practice, the exploration and exploitation contradicts to each other. In order to achieve good performances on problem optimizations, the two abilities should be well balanced. While, we observed that the solution search equation of ABC algorithm is good at exploration but poor at exploitation. Inspired by PSO [4], in this paper, we modify the solution search equation by applying the global best (gbest) solution to guide the search of new candidate solutions in order to improve the exploitation. It should be pointed out that global best solution has also been utilized by DE and harmony search in some cases [12,21]. We name the ABC algorithm using the modified solution search equation as Gbest-guided ABC (GABC) algorithm. Our experiment results tested on numerical function optimization show that GABC algorithm with appropriate parameter is superior to ABC algorithm in the most cases.

The rest of this paper is organized as follows. Section 2 summarizes ABC algorithm. The modified ABC algorithm called GABC algorithm is presented in Section 3. Section 4 presents and discusses the experimental results. Finally, the conclusion is drawn in Section 5.

2. Overview of artificial bee colony (ABC) algorithm

In a natural bee swarm, there are three kinds of honey bees to search foods generally, which include the employed bees, the onlookers, and the scouts (both the onlookers and the scouts are also called unemployed bees). The employed bees search the food around the food source in their memory, meanwhile they pass their food information to the onlookers. The onlookers tend to select good food sources from those founded by the employed bees, then further search the foods around the selected food source. The scouts are translated from a few employed bees, which abandon their food sources and search new ones. In a word, the food search of bees is collectively performed by the employed bees, the onlookers, and the scouts.

By simulating the foraging behaviors of honey bee swarm, Karaboga [7] recently invented ABC algorithm for numerical function optimization. The framework of ABC algorithm [7,9,10] can be described in Fig. 1.

There are some important details that should be pointed out for the framework of ABC algorithm described in Fig. 1. Firstly, the update process used in the onlooker stage is the same as that in the employed bee stage. Given a solution x_i to be updated (here x_i denotes the ith solution in the population), and let $v_i = x_i$. In the update process, a new candidate solution is firstly given by the following solution search equation:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \tag{1}$$

where x_{ij} (or v_{ij}) denotes the jth element of x_i (or v_i), and j is a random index. x_k denotes another solution selected randomly from the population. And ϕ_{ij} is a uniform random number in [-1, 1]. Then, a greedy selection is done between x_i and v_i, which completes the update process. Secondly, in the onlooker stage, the solutions are selected according to the probability $P_i = \mathrm{fit}_i / \sum_n \mathrm{fit}_n$, where fit_i denotes the fitness value of the ith solution in the population. Thirdly, the main distinction between the employed bee stage and the onlooker stage is that every solution in the employed bee stage involves the update process, while only the selected solutions have the opportunity to update in the onlooker stage. Fourthly, an inactive solution of the scout stage refers to a solution that does not change over a certain number of generations.

3. Gbest-guided ABC (GABC) algorithm

As well known that both exploration and exploitation are necessary for the population-based optimization algorithms, such as GA [1,2], PSO [3,4], DE [12], and so on. In these optimization algorithms, the exploration refers to the ability to investigate the various unknown regions in the solution space to discover the global optimum. While, the exploitation refers to the ability to apply the knowledge of the previous good solutions to find better solutions [22]. In practice, the exploration

Initialization
Repeat
- Employed bee stage: Perform an update process for each solution in the solution population.
- Onlooker stage: Randomly select solutions depending on their fitness values, then perform the same update process for each selected solution.
- Scout stage: Select one of the most inactive solutions, then replace it by a new randomly generated solution.

Until (conditions are satisfied)

Fig. 1. Framework of ABC algorithm.

and exploitation contradict with each other, and in order to achieve good optimization performance, the two abilities should be well balanced. According to the solution search equation of ABC algorithm described by Eq. (1), the new candidate solution is generated by moving the old solution towards (or away from) another solution selected randomly from the population. However, the probability that the randomly selected solution is a good solution is the same as that the randomly selected solution is a bad one, so the new candidate solution is not promising to be a solution better than the previous one. On the other hand, in Eq. (1), the coefficient ϕ_{ij} is a uniform random number in [-1,1] and x_{kj} is a random individual in the population, therefore, the solution search dominated by Eq. (1) is random enough for exploration. To sum up, the solution search equation described by Eq. (1) is good at exploration but poor at exploitation.

Inspired by PSO [4], which, in order to improve the exploitation, takes advantage of the information of the global best (gbest) solution to guide the search of candidate solutions, we modify the solution search equation described by Eq. (1) as follows

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{k,j}) + \psi_{ij}(y_j - x_{i,j}), \tag{2}$$

where the third term in the right-hand side of Eq. (2) is a new added term called gbest term, y_j is the jth element of the global best solution, ψ_{ij} is a uniform random number in $[0, C]$, where C is a nonnegative constant. According to Eq. (2), the gbest term can drive the new candidate solution towards the global best solution, therefore, the modified solution search equation described by Eq. (2) can increase the exploitation of ABC algorithm. Note that the parameter C in Eq. (2) plays an important role in balancing the exploration and exploitation of the candidate solution search. When C takes 0, Eq. (2) is identical to Eq. (1). When C increases from zero to a certain value,[1] the exploitation of Eq. (2) will also increase correspondingly. However, C should not be too large because of two reasons. One is that the large value of C might result in relatively weakening the exploration of Eq. (2). The second reason is that if C takes a large value, according to Eq. (2) again, the gbest term likely drives the new candidate solution move over the global best solution, which will also weaken the exploitation of Eq. (2).

In this paper, we modify ABC algorithm by replacing Eq. (1) with Eq. (2), and name the modified ABC algorithm as Gbest-guided ABC (GABC) algorithm. Although the solution search equation of GABC algorithm described by Eq. (2) is similar to those of DE [12] and PSO [3,4], GABC algorithm still preserves the main characteristics of ABC algorithm, which can distinguish GABC algorithm from DE and PSO. GABC algorithm is clearly different from PSO, because that, like DE, GABC algorithm does a comparison between the new candidate solution and the old solution, and then just saves the better one, while PSO does not involve such selection procedure. The employed bee stage of GABC algorithm has much in common with the DE that does not include the crossover stage. However, the whole of GABC algorithm consists of the three different stages that are the employed bee stage, onlooker stage and the scout stage. As mentioned above, the onlooker stage tends to select the good solution to further update, while both the employed bee stage and DE update every individual in the population. Furthermore, the scout stage is a peculiar stage of ABC algorithm, which discards an inactive solution and then randomly generates a new solution to replace the discarded one.

4. Experiments

4.1. Benchmark functions

In order to test the performance of GABC algorithm on numerical function optimization, six numerical benchmark functions used in [9,10] are used here.

The first function is the generalized Schaffer function described by

$$f_1(\vec{x}) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^{D} x_i^2}\right) - 0.5}{\left(1 + 0.001\left(\sum_{i=1}^{D} x_i^2\right)\right)^2}, \tag{3}$$

where $\vec{x} = [x_1, x_2, \ldots, x_D]$, the initial range of $\vec{x}$ is $[-100, 100]^D$, and D denotes the dimension of the solution space. The minimum solution of the Schaffer function is $\vec{x}^* = [0, 0, \ldots, 0]$, and $f_1(\vec{x}^*) = 0$.

The second function is the Rosenbrock function described by

$$f_2(\vec{x}) = \sum_{i=1}^{D-1} 100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2, \tag{4}$$

where the initial range of $\vec{x}$ is $[-50, 50]^D$. The minimum solution of the Rosenbrock function is $\vec{x}^* = [1, 1, \ldots, 1]$, and $f_2(\vec{x}^*) = 0$.

The third function is the Sphere function described by

$$f_3(\vec{x}) = \sum_{i=1}^{D} x_i^2, \tag{5}$$

[1] When C takes 2, the mean of the random number ψ_{ij} is 1. By experience, the certain value is 2.

where the initial range of $\vec{x}$ is $[-100,100]^D$. The minimum solution of the Sphere function is $\vec{x}^* = [0,0,\ldots,0]$, and $f_3(\vec{x}^*) = 0$.

The fourth function is the Griewank function described by

$$f_4(\vec{x}) = \frac{1}{4000}\left(\sum_{i=1}^{D}(x_i - 100)^2\right) - \left(\prod_{i=1}^{D}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right)\right) + 1, \tag{6}$$

where the initial range of $\vec{x}$ is $[-600,600]^D$. The minimum solution of the Griewank function is $\vec{x}^* = [100,100,\ldots,100]$, and $f_4(\vec{x}^*) = 0$.

The fifth function is the Rastrigin function described by

$$f_5(\vec{x}) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right), \tag{7}$$

where the initial range of $\vec{x}$ is $[-5.12,5.12]^D$. The minimum solution of the Rastrigin function is $\vec{x}^* = [0,0,\ldots,0]$, and $f_5(\vec{x}^*) = 0$.

The sixth function is the Ackley function described by

$$f_6(\vec{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right), \tag{8}$$

where the initial range of $\vec{x}$ is $[-32.768,32.768]^D$. The minimum solution of the Ackley function is $\vec{x}^* = [0,0,\ldots,0]$, and $f_6(\vec{x}^*) = 0$.

4.2. Parameter Settings

Some comparative experiments on numerical function optimization have been conducted for ABC algorithm in [7,9–11]. The experimental results show that ABC algorithm is competitive with some conventional optimization algorithms, such as GA [1,2], DE [12], and PSO [3,4]. In this section, a set of experiments tested on six numerical benchmark function were performed to compare the performance of GABC algorithm with that of ABC algorithm. And in order to investigate the effect of the parameter C of the solution search equation described by Eq. (2) on the performance of GABC algorithm, GABC algorithm was tested with C = 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, respectively. Note that ABC algorithm is a special case of GABC algorithm, that is, ABC algorithm is identical to GABC algorithm with C = 0. For each benchmark function, two kinds of dimensions of solution space were tested. Both the Schaffer function and the Rosenbrock function were tested with dimensions 2 and 3, while the Sphere function, the Griewank function, the Rastrigin function and the Ackley function were all tested with dimensions 30 and 60. All experiments were run for 400,000 function evaluations (the population size is 80 and the maximum number of generations is 5000) or until the function error dropped below e-20 (values less than e-20 were reported as 0). Each of the experiments was repeated 30 times independently. And the reported results are the means and standard deviations of the statistical experimental data.

4.3. Experimental results

Tables 1–6 shows the optimization results of the Schaffer function, the Rosenbrock function, the Sphere function, the Griewank function, the Rastrigin function and the Ackley function, respectively. As mentioned in Section 3, the parameter C of GABC algorithm plays an important role in controlling the exploration and exploitation of the new candidate solution search. Roughly speaking, when the parameter C increases from zero to a certain value, the exploitation of GABC algorithm enhances, while the exploration decreases relatively. It can be observed that for each of the six function optimizations, with the increase of the value of the parameter C, the mean beast value of the optimized function firstly decreases (which means

Table 1
Optimizations of the Schaffer function by ABC algorithm and GABC algorithm with C = 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, respectively. The bold values are the minimum values in each "Mean" column.

Algorithm	Schaffer function (f_1)			
	D = 2		D = 3	
	Mean	SD	Mean	SD
ABC	**0**	0	5.767849e−06	1.615e−05
GABC (C = 0.5)	**0**	0	2.203874e−09	1.207e−08
GABC (C = 1.0)	**0**	0	9.251858e−18	2.104e−17
GABC (C = 1.5)	**0**	0	**1.850371e−18**	1.013e−17
GABC (C = 2.0)	**0**	0	5.551115e−18	1.693e−17
GABC (C = 3.0)	**0**	0	1.135882e−12	5.434e−12
GABC (C = 4.0)	**0**	0	3.416302e−06	1.000e−05

Table 2
Optimizations of the Rosenbrock function by ABC algorithm and GABC algorithm with $C = 0.5, 1.0, 1.5, 2.0, 3.0, 4.0$, respectively. The bold values are the minimum values in each "Mean" column.

Algorithm	Rosenbrock function (f_2)			
	$D = 2$		$D = 3$	
	Mean	SD	Mean	SD
ABC	9.931357e−03	8.143e−03	6.449468e−02	4.852e−02
GABC ($C = 0.5$)	1.092059e−03	1.128e−03	9.064147e−03	9.036e−03
GABC ($C = 1.0$)	3.937204e−04	4.533e−04	**2.635891e−03**	2.115e−03
GABC ($C = 1.5$)	**1.684969e−04**	1.454e−04	2.659139e−03	2.220e−03
GABC ($C = 2.0$)	3.295771e−04	3.112e−04	3.913793e−03	3.560e−03
GABC ($C = 3.0$)	1.418120e−03	1.517e−03	1.190091e−02	9.205e−03
GABC ($C = 4.0$)	1.735364e−03	1.906e−03	1.982736e−02	1.532e−02

Table 3
Optimizations of the Sphere function by ABC algorithm and GABC algorithm with $C = 0.5, 1.0, 1.5, 2.0, 3.0, 4.0$, respectively. The bold values are the minimum values in each "Mean" column.

Algorithm	Sphere function (f_3)			
	$D = 30$		$D = 60$	
	Mean	SD	Mean	SD
ABC	6.379110e−16	1.203e−16	2.277413e−15	3.178e−16
GABC ($C = 0.5$)	5.220121e−16	4.769e−17	1.619816e−15	1.978e−16
GABC ($C = 1.0$)	4.316764e−16	7.499e−17	1.434367e−15	1.798e−16
GABC ($C = 1.5$)	**4.176106e−16**	7.365e−17	**1.433867e−15**	1.375e−16
GABC ($C = 2.0$)	4.390664e−16	6.675e−17	1.459731e−15	1.375e−16
GABC ($C = 3.0$)	5.211785e−16	7.260e−17	1.792182e−15	2.993e−16
GABC ($C = 4.0$)	6.040497e−16	1.211e−16	2.133957e−15	3.857e−16

Table 4
Optimizations of the Griewank function by ABC algorithm and GABC algorithm with $C = 0.5, 1.0, 1.5, 2.0, 3.0, 4.0$, respectively. The bold values are the minimum values in each "Mean" column.

Algorithm	Griewank function (f_4)			
	$D = 30$		$D = 60$	
	Mean	SD	Mean	SD
ABC	1.273055e−15	1.464e−15	2.510399e−13	7.514e−13
GABC ($C = 0.5$)	2.072416e−16	1.767e−16	1.532107e−15	1.352e−15
GABC ($C = 1.0$)	8.881784e−17	8.450e−17	9.473903e−16	7.849e−16
GABC ($C = 1.5$)	**2.960594e−17**	4.993e−17	**7.549516e−16**	4.127e−16
GABC ($C = 2.0$)	8.141635e−17	9.189e−17	9.103828e−16	4.367e−16
GABC ($C = 3.0$)	4.770665e−13	2.605e−12	6.509237e−14	2.930e−13
GABC ($C = 4.0$)	2.065754e−14	6.152e−15	3.947221e−08	1.961e−07

Table 5
Optimizations of the Rastrigin function by ABC algorithm and GABC algorithm with $C = 0.5, 1.0, 1.5, 2.0, 3.0, 4.0$, respectively. The bold values are the minimum values in each "Mean" column.

Algorithm	Rastrigin function (f_5)			
	$D = 30$		$D = 60$	
	Mean	SD	Mean	SD
ABC	1.345294e−13	7.966e−14	2.064794e−08	1.121e−07
GABC ($C = 0.5$)	3.979039e−14	2.649e−14	7.124375e−13	3.752e−13
GABC ($C = 1.0$)	**9.473903e−15**	2.154e−14	4.168517e−13	1.774e−13
GABC ($C = 1.5$)	1.326346e−14	2.445e−14	**3.524291e−13**	1.243e−13
GABC ($C = 2.0$)	1.515824e−14	2.556e−14	5.078012e−13	3.701e−13
GABC ($C = 3.0$)	5.684341e−14	3.949e−14	1.272534e−11	3.264e−11
GABC ($C = 4.0$)	1.004233e−13	6.277e−14	6.996970e−09	2.238e−08

the solution gets better), and then begins to increase (which means the solution gets worse) at a certain point in the most cases, two of which are illustrated in Figs. 2 and 3. It can be also observed that the performances of GABC algorithm with $C = 0.5, 1.0, 1.5$ are all superior to ABC algorithm. And, as a whole, GABC algorithm with $C = 1.5$ has the best performance among the tested algorithms. By synthesizing the data in Tables 1–6, we made Table 7, which can clearly show the comparison between GABC algorithm with $C = 1.5$ and ABC algorithm on optimizing the six benchmark functions. We have drawn

Table 6
Optimizations of the Ackley function by ABC algorithm and GABC algorithm with C = 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, respectively. The bold values are the minimum values in each "Mean" column.

Algorithm	Ackley (f_6)			
	D = 30		D = 60	
	Mean	SD	Mean	SD
ABC	4.695503e−14	5.954e−15	1.660893e−13	2.217e−14
GABC (C = 0.5)	3.961275e−14	3.564e−15	1.193119e−13	9.049e−15
GABC (C = 1.0)	3.309944e−14	2.903e−15	1.039168e−13	1.079e−14
GABC (C = 1.5)	**3.215205e−14**	3.252e−15	**1.000088e−13**	6.089e−15
GABC (C = 2.0)	3.309944e−14	3.694e−15	1.003641e−13	7.028e−15
GABC (C = 3.0)	3.854694e−14	3.278e−15	1.183645e−13	9.771e−15
GABC (C = 4.0)	4.411286e−14	5.776e−15	1.626550e−13	2.542e−14

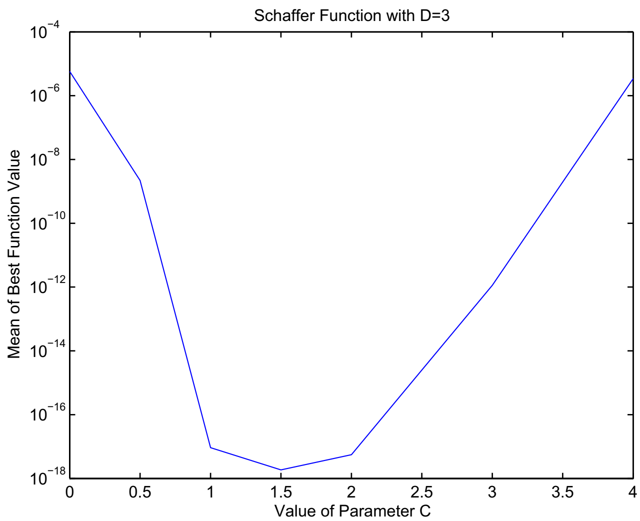

Fig. 2. The variation of the mean best value of the Schaffer function (D = 3) with the change of the parameter C of GABC algorithm.

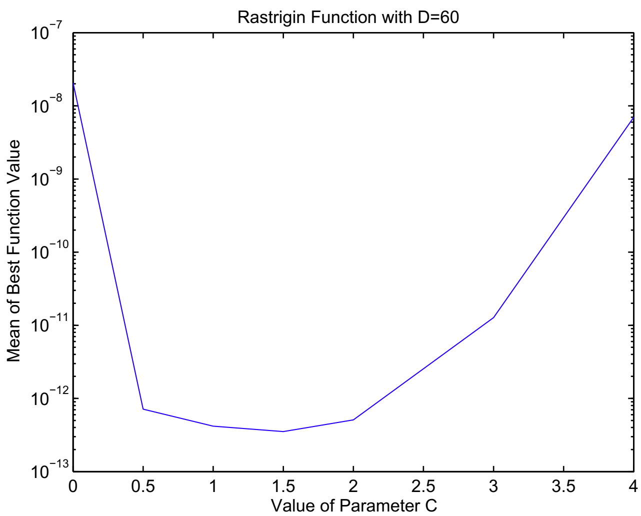

Fig. 3. The variation of the mean best value of the Rastrigin function (D = 60) with the change of the parameter C of GABC algorithm.

the convergence curves of ABC and GABC algorithms to show the progresses of the mean best values presented in Table 7. For space limitation, here we just present two representative cases of the convergence curves, which are shown in Figs. 4 and 5,

Table 7
Comparison between GABC algorithm ($C = 1.5$) and ABC algorithm on optimizing six benchmark functions. The bold values are the minimum values between "ABC-Mean" column and "GABC-Mean" column.

Function	D	ABC		GABC ($C = 1.5$)	
		Mean	SD	Mean	SD
f_1	2	**0**	0	**0**	0
	3	5.767849e−06	1.615e−05	**1.850371e−18**	1.013e−17
f_2	2	9.931357e−03	8.143e−03	**1.684969e−04**	1.454e−04
	3	6.449468e−02	4.852e−02	**2.659139e−03**	2.220e−03
f_3	30	6.379110e−16	1.203e−16	**4.176106e−16**	7.365e−17
	60	2.277413e−15	3.178e−16	**1.433867e−15**	1.375e−16
f_4	30	1.273055e−15	1.464e−15	**2.960594e−17**	4.993e−17
	60	2.510399e−13	7.514e−13	**7.549516e−16**	4.127e−16
f_5	30	1.345294e−13	7.966e−14	**1.326346e−14**	2.445e−14
	60	2.064794e−08	1.121e−07	**3.524291e−13**	1.243e−13
f_6	30	4.695503e−14	5.954e−15	**3.215205e−14**	3.252e−15
	60	1.660893e−13	2.217e−14	**1.000088e−13**	6.089e−15

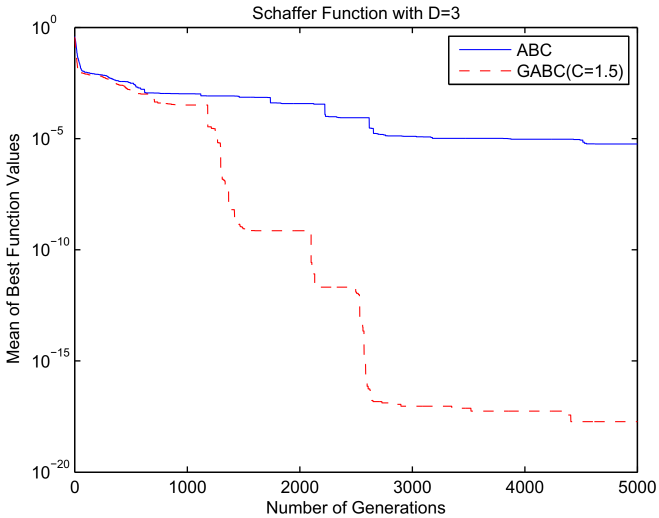

Fig. 4. Convergence curves of ABC and GABC ($C = 1.5$) algorithms for the Schaffer function ($D = 3$).

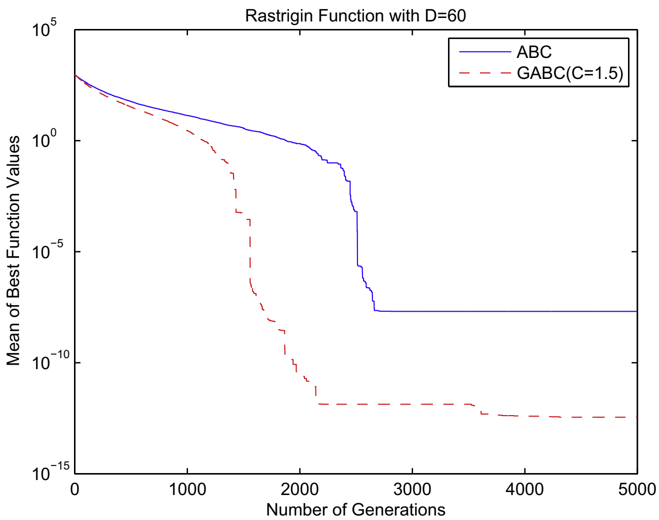

Fig. 5. Convergence curves of ABC and GABC ($C = 1.5$) algorithms for the Rastrigin function ($D = 60$).

respectively. Table 7 and the convergence curves show that GABC algorithm with $C = 1.5$ outperforms ABC algorithm in most of the experiments.

5. Conclusion

In this paper, artificial bee colony (ABC) algorithm was studied. Observing that the solution search equation of ABC algorithm is good at exploration but poor at exploitation, we proposed an improved ABC algorithm called Gbest-guided ABC (GABC) algorithm, which takes advantage of the information of global best solution to guide the search of new candidate solutions in order to improve the exploitation. The experimental results tested on six benchmark functions show that GABC algorithm with appropriate parameter outperforms ABC algorithm.

Acknowledgement

The authors thank the anonymous reviewers for their valuable comments and suggestions. This work is partly supported by Hong Kong RGC General Research Fund (GRF) 9041495 (CityU 115109). This work was also supported in part by the NSFC (Grant No. 61003297, 40701050).

References

[1] J. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, 1992.
[2] K.S. Tang, K.F. Man, S. Kwong, Q. He, Genetic algorithms and their applications, IEEE Signal Processing Magazine 13 (1996) 22–37.
[3] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[4] R. Eberhart, Y. Shi, J. Kennedy, Swarm Intelligence, Morgan Kaufmann, San Fransisco, CA, 2001.
[5] M. Dorigo, T. Stutzle, Ant Colony Optimization, MIT Press, Cambridge, MA, 2004.
[6] D. Simon, Biogeography-based optimization, IEEE Transactions on Evolutionary Computation 12 (2008) 702–713.
[7] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Erciyes University, Kayseri, Turkey, Technical Report-TR06, 2005.
[8] B. Basturk, D. Karaboga, An artificial bee colony (ABC) algorithm for numeric function optimization, in: IEEE Swarm Intelligence Symposium, 2006.
[9] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization 39 (2007) 171–459.
[10] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing 8 (2008) 687–697.
[11] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation 214 (2009) 108–132.
[12] K.V. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin, Germany, 2005.
[13] A. Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, Applied Soft Computing 9 (2009) 625–631.
[14] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, Journal of The Franklin Institute 346 (2009) 328–348.
[15] J.W. Ponton, J. Klemes, Alternatives to neural networks for inferential measurement, Computers and Chemical Engineering 17 (1993) 42–47.
[16] R.S. Rao, S. Narasimham, M. Ramalingaraju, Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm, International Journal of Electrical Power and Energy Systems Engineering (IJEPESE) 1 (2008) 116–122.
[17] D. Karaboga, B. Akay, C. Ozturk, Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, LNCS: Modeling Decisions for Artificial Intelligence, vol. 4617, Springer-Verlag, 2007, pp. 318–329.
[18] P. Pawar, R. Rao, J. Davim, Optimization of process parameters of milling process using particle swarm optimization and artificial bee colony algorithm, in: International Conference on Advances in Mechanical Engineering, 2008.
[19] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, LNCS: Advances in Soft Computing-Foundations of Fuzzy Logic and Soft Computing, vol. 4529, Springer-Verlag, 2007, pp. 789–798.
[20] Q.-K. Pan, M.F. Tasgetiren, P.N. Suganthan, T.J. Chua, A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, Information Sciences, in press.
[21] M.G.H. Omran, M. Mahdavi, Global-best harmony search, Applied Mathematics and Computation 198 (2008) 643–656.
[22] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (2003) 317–325.