Hindawi Scientific Programming Volume 2018, Article ID 2767546, 9 pages https://doi.org/10.1155/2018/2767546



# Research Article

# **An Artificial Bee Colony Algorithm with Random Location Updating**

# Lijun Sun (b), <sup>1</sup> Tianfei Chen (b), <sup>1</sup> and Qiuwen Zhang (b)<sup>2</sup>

<sup>1</sup>School of Electrical Engineering, Henan University of Technology, Zhengzhou, Henan 450001, China <sup>2</sup>College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan 450002, China

Correspondence should be addressed to Tianfei Chen; chen\_tianfei@163.com

Received 8 April 2018; Accepted 11 July 2018; Published 1 August 2018

Academic Editor: Ricardo Soto

Copyright © 2018 Lijun Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a novel swarm intelligence algorithm, artificial bee colony (ABC) algorithm inspired by individual division of labor and information exchange during the process of honey collection has advantage of simple structure, less control parameters, and excellent performance characteristics and can be applied to neural network, parameter optimization, and so on. In order to further improve the exploration ability of ABC, an artificial bee colony algorithm with random location updating (RABC) is proposed in this paper, and the modified search equation takes a random location in swarm as a search center, which can expand the search range of new solution. In addition, the chaos is used to initialize the swarm population, and diversity of initial population is improved. Then, the tournament selection strategy is adopted to maintain the population diversity in the evolutionary process. Through the simulation experiment on a suite of unconstrained benchmark functions, the results show that the proposed algorithm not only has stronger exploration ability but also has better effect on convergence speed and optimization precision, and it can keep good robustness and validity with the increase of dimension.

#### 1. Introduction

Many important problems require optimization, including the traveling salesman problem, job shop scheduling, and neural network training [1]. The existing algorithms such as genetic algorithm (GA), evolutionary computation (EC), and particle swarm optimization (PSO) are the simulation of intelligence from the perspective of biological evolution to solve optimization problems [2]. Artificial bee colony (ABC) algorithm is a new kind of swarm intelligence algorithm proposed by Karaboga and Basturk [3]; it simulates the intelligent behavior of honey bees, and bees carry out different nectar collecting activities according to their respective division of labor to realize the sharing and exchange of information source. Because of its simple structure, less parameters, and easy implementation, it has received extensive attention and research from many

scholars. At present, ABC algorithm has been successfully applied in many fields, such as neural network [4], filter design [5], parameter optimization [6], and combinatorial optimization [7]. Similar to other swarm intelligence algorithms, the ABC algorithm also suffers from defect of early maturity. For this reason, some scholars use chaos to initialize population to improve the diversity and the ergodicity of swarm [8, 9]. In [10, 11], the selection mechanism of bidding match and ranking is put forward to reduce the influence of super-individuals in the population so as to avoid early maturity. In order to make the algorithm jump out of a local extremum, the mutation operators are usually integrated into the ABC algorithm. The commonly used mutation operators include Gauss mutation [12], Cauchy mutation [13], and differential evolution mutation [14]. Rajasekhar et al. [15] makes full use of Lévy distribution which has normal distribution and Cauchy

distribution characteristics, and an improved ABC algorithm based on Lévy mutation is put forward.

As to further improve the performance of ABC algorithm, some scholars integrate the ABC algorithm and other intelligent optimization algorithms, and some improved hybrid optimization algorithms are proposed. Kıran and Gündüz [16] proposed a hybrid optimization algorithm based on particle swarm optimization (PSO) and ABC algorithm. A hybrid algorithm combining ABC algorithm and quantum evolution is presented by Duan to solve the continuous optimization problem [17]. Chen et al. [18] integrated a simulated annealing operator to ABC algorithm, which can improve the exploitation ability of the algorithm. Although ABC algorithm has better exploration ability, the exploitation ability is insufficient and the local search ability is weak. Zhu and Kwong [19] proposed a global best- (gbest-) guided ABC by incorporating the information of the gbest solution into the solution search equation of ABC to improve the exploitation. Because the Nelder-Mead simplex search (NMSS) mechanism is a local descent algorithm, Kang et al. [20] combined NMSS and ABC, and a hybrid simplex ABC algorithm is presented. The Rosenbrock ABC algorithm is proposed, which takes the Rosenbrock method as a local exploitation operator [21]. Gao et al. [22] used the traditional Powell method as a local search operator and combined it with ABC, which can complement the exploration ability of the Powell method and the exploitation ability of ABC, thus the performance of the algorithm is improved.

In this paper, an artificial bee colony algorithm with random location updating (RABC) is proposed, the basic search equation is modified, which can expand the search range of new solution and further improve the exploration ability of ABC algorithm. Numerical simulation experiments are carried on some benchmark functions, and the results demonstrate the effectiveness of RABC. The rest of this paper is organized as follows. In Section 2, the basic ABC algorithm is presented. In Section 3, the RABC algorithm is introduced in detail. Section 4 presents and discusses the experimental results. Finally, the conclusion is drawn is Section 5.

# 2. ABC Algorithm

2.1. Behavior Description of Bee Swarm. Bees in nature have three different roles while gathering nectar; they are, respectively, employed bees, onlooker bees, and scout bees. The three kinds of bees cooperate with each other to accomplish nectar collection. In addition, the basic behavior of bees can be divided into two categories, which are the recruitment of bees for food sources and the abandonment of a food source.

Food sources are equivalent to locations of solution in the optimization problem, and the quality of the food source is expressed by a fitness value. The primary task of the employed bees and the scout bees is to explore and exploit food sources. After watching the waggle dance of employed bees, onlooker bees determine the yield of the food source through the intensity and

duration of the waggle dance and then choose the food source to collect nectar based on the yield, and the scout bees are randomly searching for food sources near the hive. The bees that have already found food sources are called the employed bees. The employed bees store some information about food sources; they match food sources one to one and share the information with other bees at certain probability. The number of the employed bees is the same as that of the onlooker bees, and the scout bee is only one.

The search process for food sources consists of three steps: (1) the employed bees find food sources and record the amount of nectar; (2) according to the nectar information provided by the employed bees, the onlooker bees select the food source to collect nectar; and (3) the scout bees are determined to search for new food sources.

2.2. Description of ABC. Inspired by the bee swarm intelligent behavior of gathering nectar in nature, ABC algorithm is proposed by Karaboga. In the ABC algorithm, the complete search range of bee swarm represents the solution space of optimization problem. A location of a random food source corresponds to a stochastic solution of the optimization problem, and the nectar quantity of the food source represents the fitness value, which is used to describe quality of solutions. Therefore, the gathering nectar process is also the process of searching for the optimal solution.

Considering the global optimization problem min  $\{f(\mathbf{X}): \mathbf{X} \in \mathbf{S} \subset \mathbf{R}^n\}$ , the size of food sources is set as SN, and the food source  $\mathbf{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$  represents a candidate solution;  $i \in \{1, 2, \dots, \text{SN}\}$ , and D is the dimension of the optimization problem. First, the ABC algorithm generates an initial population containing SN solutions; the bee swarm searches all food source circularly, and the number of cycles is  $C(C = 1, 2, \dots, \text{MCN})$ . Eventually, the optimal solution can be found. The search process includes three phases, which are depicted in detail as follows:

#### (i) Employed Bees Phase

In this phase, each employed bee corresponds to the current food source  $\mathbf{X}_i$ , and a new food source  $\mathbf{V}_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,D})$  can be generated by using the search equation; the search equation is expressed as follows:

$$v_{i,j} = x_{i,j} + \varphi_{i,j} (x_{i,j} - x_{k,j}), \tag{1}$$

where  $\varphi_{i,j}$  is a uniform random number in [-1, 1] and  $k \in \{1, 2, \ldots, SN\}$  is the randomly chosen indices and  $k \neq i$ .  $j \in \{1, 2, \ldots, D\}$  indicates a random selection dimension. If the fitness of the new solution  $\mathbf{V}_i$  is better, the original solution is replaced by the new solution; otherwise, the original solution will be retained.

#### (ii) Onlooker Bees Phase

After all the employed bees have completed the search process, they will share the information of food sources with the onlooker bees in the dance area, and the onlooker bees

will calculate the probability of each solution according to the following formula:

$$p_i = \frac{\text{fit}_i}{\sum_{k=1}^{\text{SN}} \text{fit}_k},\tag{2}$$

where fit<sub>i</sub> is the fitness value of the solution i. A random number is generated in [0, 1]. If the probability of the solution  $p_i$  is greater than the random number, then the onlooker bee will generate a new solution by (1), and the fitness value of the new solution is calculated. If the new solution is better, the original solution is replaced; otherwise, the original solution is reserved. It can be seen that the better the fitness of the food source is, the higher the probability onlooker bees select.

#### (iii) Scout Bees Phase

If a position cannot be improved further through a predetermined number of cycles *limit*, it indicates that the food source has been exhausted and then that food source is assumed to be abandoned. The current employed bee becomes the scout bee, and the scout bee will produce a new food source to replace the old one according to following formula:

$$x_{i,j} = x_{\min,j} + \text{rand}(0,1) \cdot (x_{\max,j} - x_{\min,j}),$$
 (3)

where  $x_{\max,j}$ , and  $x_{\min,j}$  are the upper and lower bounds for the dimension j, respectively.

The above three phases will be repeated until the maximum iteration number MCN or the allowable value of error  $\varepsilon$  is achieved.

The fitness function is used to measure nectar quality of solutions in ABC algorithm, and it is selected corresponding to optimization problem. If the optimization problem is to find the minimum value, the fitness function is deformation of the objective function  $f(\cdot)$ , and it is expressed as follows:

$$\operatorname{fit}(\mathbf{X}_{i}) = \begin{cases} 1/(1+f(\mathbf{X}_{i})), & f(\mathbf{X}_{i}) > 0, \\ 1+(\operatorname{abs}(f(\mathbf{X}_{i}))), & f(\mathbf{X}_{i}) \leq 0. \end{cases}$$
(4)

If the optimization problem is to find the maximum value, the fitness function is the objective function.

The greedy selection is performed in ABC algorithm, where

$$\mathbf{V}_{i} = \begin{cases} \mathbf{V}_{i}, & \text{fit}(\mathbf{V}_{i}) > \text{fit}(\mathbf{X}_{i}), \\ \mathbf{X}_{i}, & \text{fit}(\mathbf{V}_{i}) \leq \text{fit}(\mathbf{X}_{i}). \end{cases}$$
(5)

# 3. ABC with Random Location Updating

3.1. Search Range Analysis and Modified Search Equation. In ABC, the bee swarm relies on the information sharing among individuals to explore new food sources, and the algorithm has stronger randomness, so it is generally recognized that it has better global search ability. Therefore, some scholars have carried out research on how to improve the local search ability of ABC algorithm. The representative work is the global best- (gbest-) guided ABC proposed by Zhu (GABC). It is inspired by the particle swarm algorithm,

and a search operator guided by the best solution is introduced to improve the exploit ability of the algorithm. Thus, the search equation is expressed as follow:

$$v_{i,j} = x_{i,j} + \varphi_{i,j}(x_{i,j} - x_{k,j}) + \varphi_{i,j}(g \operatorname{best}_j - x_{i,j}),$$
 (6)

where the parameter  $\varphi$  is the uniformly random number distributed in [0, C] and C is the nonnegative constant. Generally, it is set to 2.

However, the main purpose of this paper is to further improve the global search ability of ABC algorithm and accelerate the convergence speed. In order to directly analyze the location of the new food source, the search range is analyzed taking the two-dimensional plane space as an example. As shown in Figure 1, in the plane coordinate system O-XY, the coordinates of points A, B, and C are respectively (1,0), (3,1), and (0,2). We take the point A as the current food source, and according to the basic search (1) in ABC, the search range defined by the points A and B is represented by the blue area, and the search range defined by the points A and C is represented by the orange area. The two search area overlaps, and yellow indicates the overlap area. From Figure 1, although the ABC has good global search ability and the search region is relatively large, the main areas are still concentrated near the current food source, and repeated search exists in the neighborhood of current food source. Especially, in the later evolution stage of the algorithm, it is easy to fall into local extremum and lead to the premature convergence.

In order to further expand the global search ability of ABC algorithm, we modify the search equation of ABC algorithm and propose an improved ABC algorithm with random location updating. This algorithm takes a random position of population as the search center, and thus the modified search equation is expressed as

$$v_{i,j} = x_{r,j} + \varphi_{i,j} (x_{r,j} - x_{i,j}), \tag{7}$$

where  $r \in \{1, 2, ..., SN\}$  is a random selection food source and  $r \neq i$ .  $\varphi_{i,j}$  is a random number in [-1, 1]. By comparing (1) and (7), we can see that the search centers of these two equations are inconsistent. The ABC algorithm uses the current food source as the search center, while the RABC algorithm randomly selects a food source as the search center.

In order to compare and analyze the search area of RABC algorithm, the same example in Figure 1 is used to determine the search area for RABC. As shown in Figure 2, the search range defined by the points A and B is represented by the blue area similarly, and the search range defined by the points A and C is represented by the orange area. These two search areas do not overlap, and compared with Figure 1, the joint search area is expanded, and the neighbor area of the point A is partly included. Therefore, from the size of the search range, the RABC algorithm can have better global search capability and will be further validated by numerical experiments in the following section.

3.2. Chaos Initialization. Chaos is a nonlinear phenomenon and widely exists in nature. Generally, chaos is defined by

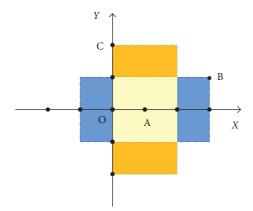


FIGURE 1: Search range of ABC.

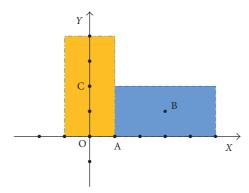


FIGURE 2: Search range of RABC.

a deterministic equation and can generate stochastic motion states. The generated chaotic sequence has the characteristics of randomness, ergodicity, and regularity, and it can traverse all the states in a certain range according to its own laws. In this paper, chaos is used to realize the initialization of population. At present, logistic is the most frequently used chaotic function, and its equation is as follows:

$$z_{n+1} = \mu \cdot z_n \cdot (1 - z_n), \quad n = 0, 1, \dots,$$
 (8)

where  $\mu$  is the control parameter. When the parameter  $\mu$  is determined, the sequence  $z_1, z_2, \ldots, z_n$  can be generated iteratively at any initial value  $z_0 \in [0, 1]$ , and if  $\mu = 4$ , the system of (6) is completely chaotic. We use logistic chaos to initialize the population; it can not only retain the randomness of initialization but also increase the diversity of the population. The initialization process is as follows:

Step 1. For the optimization problem, the initial values are randomly generated  $z_{0,j} \in [0,1]$  and  $j \in \{1,2,..., D\}$ .

*Step 2.* The initial value is substituted into (8) to produce the chaotic sequence  $z_{i,j} \in [0,1]$  and  $i \in \{1,2,\ldots, SN\}$ .

*Step 3*. The chaotic sequence is transferred into the solution space according to the following equation:

$$x_{i,j} = x_{\min,j} + z_{i,j} \cdot \left(x_{\max,j} - x_{\min,j}\right). \tag{9}$$

3.3. Tournament Selection Strategy. In the ABC algorithm, the selection probability of food source depends on the fitness value, and the food source with the higher fitness value has greater selection probability. In the early evolution stage, some super-individuals may be generated so that the evolution is easy to fall into the local extremum, and the premature convergence comes up. Tournament selection strategy is adopted based on the competition mechanism in this paper. q = 2 individuals were randomly selected from the population and compared with each other, and then the individual with greater fitness is added 1. When all the individuals have been repeated with the above process, the highest scoring individual has the biggest weight. The tournament selection strategy increases the selection probability of a poor food source and avoids the negative effects of the super-individual on the selection process. Thus, the probability of food source selection is as follows:

$$P_{i}(t) = \frac{c_{i}(t)}{\sum_{i=1}^{SN} c_{i}(t)},$$
(10)

where  $c_i$  is the score of the *i*th individual.

3.4. Algorithm Framework. The framework of RABC Algorithm 1 is given as follows. FEs indicates the current evaluation times of the fitness function, and MaxFEs is the maximum number of fitness function evaluation. trial is used to record the number of times of the food source  $X_i$  which has not been updated.

## 4. Experiment Results

4.1. Benchmark Functions and Parameter Settings. In this section, we use five typical benchmark functions to validate the comprehensive performance of the proposed algorithm, and the experiments were implemented on a computer with the main frequency of 3.6 GHz, memory 8G, Matlab R2013a, and Windows 7 operating system. Table 1 lists the details of the five typical benchmark functions, such as search range and the minimum value. The functions  $f_1$  and  $f_2$  are typically unimodal functions, and the sphere function is mainly used to test the convergence speed. While the Rosenbrock function has the local minimum value, and it is nonconvex, ill-conditioned unimodal function used to test the convergence speed and execution efficiency. The functions  $f_3 \sim f_5$  are the complex nonlinear multimodal functions with many local extreme values. They are mainly used to test the global search ability and the ability of jumping out from the local extremum to avoid premature convergence.

A series of experiments on the benchmark function are performed, and the results are compared and analyzed among ABC, GABC, and RABC; it mainly includes accuracy, convergence speed, and dimension expansion. For a fair comparison among ABCs, they are tested using the same settings of the parameters. The number of the employed bees is the same as that of the onlooker bees.

- (1) Generate SN food sources  $\{X_i | i = 1, 2, ..., SN\}$  by Section 3.2.
- (2) FEs = SN;
- (3) While  $FEs \leq MaxFEs$  do
- (4) % Employed Bees Phase
- (5) **for** i = 1 to SN **do**
- (6) Generate a new candidate solution  $V_i$  according to (7);
- (7) Update candidate solution  $V_i$  according to (5);
- (8) if  $f(\mathbf{V_i}) < f(\mathbf{X_i})$ , set  $trial_i = 0$ , else  $trial_i = trial_i + 1$ ; end
- (9) FEs = FEs + 1;
- (10) end
- (11) % Onlooker Bees Phase
- (12) **for** ii = 1 to SN **do**
- (13) Choose a food source  $X_i$  { $i \in 1, 2, ..., SN$ } form the current swarm by the roulette wheel selection mechanism;
- (14) Generate a new candidate solution  $V_i$  according to (7);
- (15) Update candidate solution V<sub>i</sub> according to (5);
- (16) if  $f(\mathbf{V_i}) < f(\mathbf{X_i})$ , set  $trial_i = 0$ , else  $trial_i = trial_i + 1$ ; end
- (17) FEs = FEs + 1;
- (18) end
- (19) % Scout Bees Phase
- (20) if  $trial_i > limit$ , replace  $X_i$  with a newly randomly candidate.
- (21) if FEs > MaxFEs, stop and output the best solution achieved so far, otherwise, go to step 3.

Algorithm 1: RABC.

TABLE 1: Typical benchmark functions.

Name	Function	Minimum	Range	Features
Sphere	${f}_1 = \sum_{i=1}^D x_i^2$	0	(-100,100)	Unimodal
Rosenbrock	$f_2 = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	0	(-50,50)	Unimodal
Rastrigin	$f_3 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	0	(-5.12,5.12)	Multimodal
Ackley	$f_4 = -20 \exp(-0.2\sqrt{(1/n)\sum_{i=1}^n x_i^2}) - \exp((1/n)\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	0	(-32.768, 32.768)	Multimodal
Griewank	$f_5 = (1/4000) \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	0	(-600,600)	Multimodal

The maximum number of function evaluation is set to 300000; thus, the maximum iterations' cycle is 5000, and the size of population SN = 60. In addition, all the benchmark functions are tested when the dimension D is 50 and 100, respectively, and the parameter  $limit = 0.1 \times SN \times D$ . In GABC, the parameter C is set to 2, and its setting refers to [19].

4.2. Result Analysis. For each benchmark function, every algorithm runs 30 times independently, and the performance of each algorithm is evaluated by calculating the best value, the worst value, the mean value, and the standard deviation (SD). Note that the SD is mainly used to evaluate the stability of convergence accuracy for each algorithm. Table 2 and 3 show the comparison results at dimensions D = 50 and D = 100, respectively.

As can be seen from Table 2 and 3, for the unimodal sphere function, the convergence accuracy and stability of RABC is better than that of GABC and ABC both at D = 50 and D = 100. However, for the complex unimodal Rosenbrock function and multimodal Rastrigin, Ackley,

and Griewank functions, when the dimension D=50, the convergence accuracies of RABC and GABC are better than that of ABC, and the convergence stability of RABC is better than that of GABC. As the dimension of the optimization problem increases, the computational complexity also increases dramatically, and thus the algorithm performance degenerates. But in the case of dimension D=100, the convergence accuracy and stability of RABC is still better than those of GABC and ABC algorithms.

Figures 3–7 are the evolution curves of all benchmark functions; the abscissa shows the number of iteration, and the ordinate indicates the mean value of objective function. As can be seen from the above figures, for the unimodal sphere function, the optimization accuracy of RABC, GABC, and ABC decreases linearly, the convergence speed of RABC is better than that of GABC, and GABC is better than ABC. For the complex unimodal Rosenbrock function, when the dimension D=50, the convergence speeds of RABC and GABC are much better than that of ABC in the early stage of evolution. However, because of the characteristics of the function, it is easy to fall into the local extremum and the

TABLE 2.	Recult	comparisons	of ARCs	at D = 50
IABLE Z:	Resuit	Companisons	OI ADCS	at $D = 30$ .

Function	Algorithm	Best	Worst	Mean	SD
	ABC	1.3343 <i>e</i> – 07	5.5490e – 07	2.9340 <i>e</i> – 07	1.2284e - 07
$f_1$ sphere	GABC	8.0766e - 23	7.5958e - 18	5.1502e - 19	1.7824e - 18
	RABC	1.5569e - 49	2.4871e - 45	3.0262e - 46	5.5547e - 46
	ABC	6.7942e + 06	2.3206e + 07	1.6657e + 07	3.3657e + 06
f <sub>2</sub> Rosenbrock	GABC	2.5497e + 01	1.6969e + 02	7.1468e + 01	3.9955e + 01
	RABC	3.4317e + 01	1.0555e + 02	5.6988e + 01	2.5748e + 01
	ABC	3.3160e + 02	4.2417e + 02	3.9589e + 02	1.9607e + 01
f <sub>3</sub> Rastrigin	GABC	4.5768e + 01	4.4127e + 02	1.2709e + 02	1.0648e + 02
	RABC	5.0743e + 01	9.9496e + 01	7.5982e + 01	1.4424e + 01
	ABC	6.7080e – 05	1.6450e – 04	1.0683 <i>e</i> – 04	2.8482e - 05
f <sub>4</sub> Ackley	GABC	7.0690e - 12	6.7093e - 10	1.5803e - 10	1.6938e - 10
	RABC	6.2172e - 15	1.6254e - 13	3.7718e - 14	3.2718e - 14
	ABC	1.4731e – 07	9.4283 <i>e</i> – 07	5.7262 <i>e</i> – 07	2.0506e - 07
f <sub>5</sub> Griewank	GABC	0.00e + 00	0.00e + 00	0.00e + 00	0.00e + 00
-	RABC	0.00e + 00	0.00e + 00	0.00e + 00	0.00e + 00

Table 3: Result comparisons of ABCs at D = 100.

Function	Algorithm	Best	Worst	Mean	SD
	ABC	1.9289e + 04	3.0586e + 04	2.5029e + 04	2.4108e + 03
$f_1$ sphere	GABC	8.3306e + 00	3.3362e + 02	9.2300e + 01	8.0450e + 01
	RABC	5.5639e - 13	1.4283e - 10	2.3396e - 11	3.3360e - 11
	ABC	5.7439e + 08	9.1878e + 08	7.6453e + 08	8.3811e + 07
f <sub>2</sub> Rosenbrock	GABC	1.6008e + 04	8.2963e + 05	2.5306e + 05	2.3158e + 05
•	RABC	7.5552e + 01	2.9737e + 02	1.8136e + 02	5.9380e + 01
	ABC	8.1959e + 02	1.0394e + 03	9.9539e + 02	4.3577e+01
f <sub>3</sub> Rastrigin	GABC	2.8785e + 02	1.0569e + 03	5.2517e + 02	2.0834e + 02
	RABC	2.0695e + 02	3.2237e + 02	2.5974e + 02	3.2072e + 01
	ABC	1.3725e + 01	1.5174e + 01	1.4491e + 01	4.0077e - 01
f <sub>4</sub> Ackley	GABC	1.5230e + 00	5.8137e + 00	3.0739e + 00	8.9919e - 01
	RABC	1.5018e - 07	1.2507e - 06	5.4137e - 07	3.2736e - 07
	ABC	1.5337e + 02	2.1861e + 02	1.8835e + 02	1.6630e + 01
f <sub>5</sub> Griewank	GABC	9.7595e - 01	2.8826e + 00	1.3889e + 00	3.9773e - 01
-	RABC	6.3483e - 13	3.1555e - 11	8.8974e - 12	9.3624e - 12

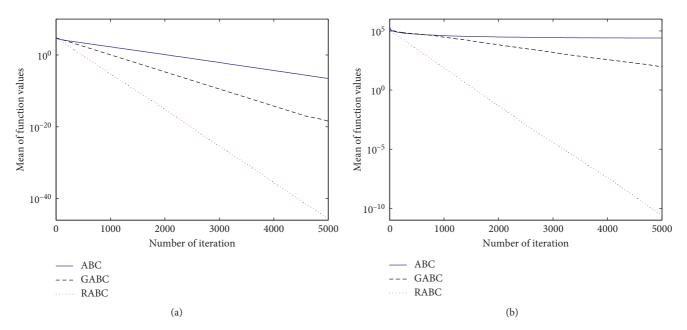


Figure 3: Evolution curves at the sphere function. (a) D = 50. (b) D = 100.

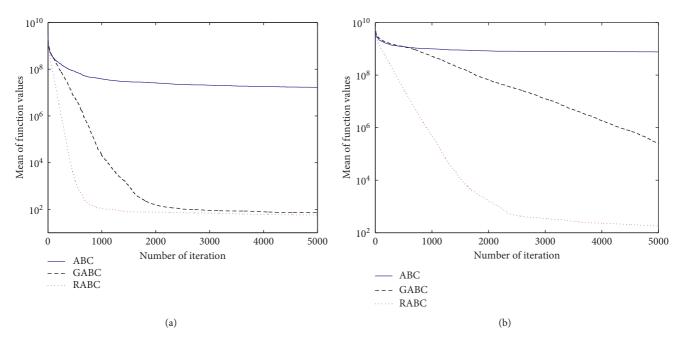


Figure 4: Evolution curves at the Rosenbrock function. (a) D = 50. (b) D = 100.

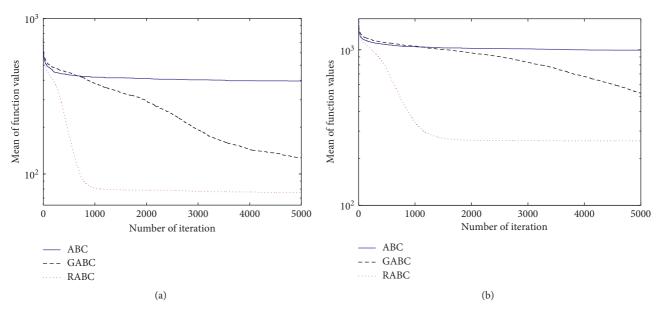


FIGURE 5: Evolution curves at the Rastrigin function. (a) D = 50. (b) D = 100.

evolution is stagnant. But in the case of dimension  $D\!=\!100$ , the convergence speed of GABC decreases significantly, while RABC still maintains a fast convergence speed. The multimodal Rastrigin, Ackley, and Griewank functions are complex nonlinear problems, which are mainly used to test the global search performance. RABC has better global search performance and fast convergence speed as shown in Figures 5–7.

In summary, whether for unimodal or multimodal functions, RABC not only has stronger exploration ability, but also has better effect on convergence speed and

optimization precision. As the dimension of the optimization problem increases, it can also keep good robustness and validity.

#### 5. Conclusions

This paper presents an artificial bee colony algorithm with random location updating, and the search equation of this algorithm takes a random position in swarm population as the search center. In contrast to ABC, the search range of new solution is further expanded, which can enhance the

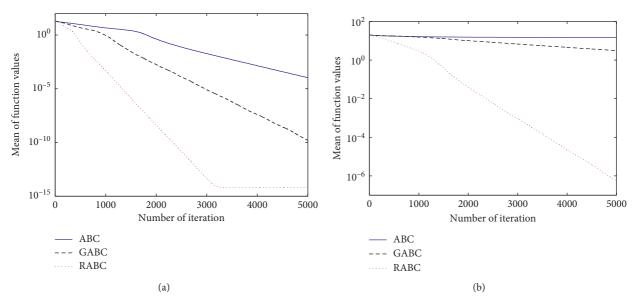


FIGURE 6: Evolution curves at the Ackley function. (a) D = 50. (b) D = 100.

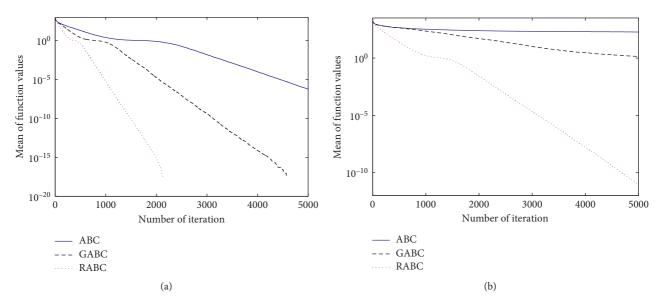


Figure 7: Evolution curves at the Griewank function. (a) D = 50. (b) D = 100.

exploration ability. Besides this, the chaos is used to initialize the swarm population, and diversity of initial population is improved. Then the tournament selection strategy is adopted to maintain the population diversity in the evolutionary process. The results of the simulation experiment on a suite of unconstrained benchmark functions demonstrate that RABC not only has stronger exploration ability but also has better effect on convergence speed and optimization precision, and it can keep good robustness and validity with the increase of dimension.

As an extension of this paper, the proposed algorithm will be further studied in theory; for instance, the global convergence of this algorithm will be verified using the convergence criterion of stochastic search algorithm and the properties of the Markov chain, and the moving trajectory of

bees will also be studied according to the theory of nonlinear dynamic. In addition, from the perspective of practical application, the fusion of the proposed algorithm with other optimization algorithms and the selection of its own parameters will also be the next research contents.

#### **Data Availability**

The data used to support the findings of this study are available from the corresponding author upon request.

## **Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

# Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (U1604151), Outstanding Talent Project of Science and Technology Innovation in Henan Province (174200510008), Program for Scientific and Technological Innovation Team in the Universities of Henan Province (16IRTSTHN029), Science and Technology Project of Henan Province (182102210094), Natural Science Project of the Education Department of Henan Province (18A510001), and the Fundamental Research funds of the Henan University of Technology (2015QNJH13 and 2016XTCX06).

## References

- [1] Y. Li, Z. H. Zhan, S. Lin, J. Zhang, and X. Luo, "Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems," *Information Sciences*, vol. 293, no. 3, pp. 370–382, 2015.
- [2] S. Strasser, J. Sheppard, N. Fortier, and R. Goodman, "Factored evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 281–293, 2017.
- [3] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [4] W. C. Yeh and T. J. Hsieh, "Artificial bee colony algorithmneural networks for S-system models of biochemical networks approximation," *Neural Computing and Applications*, vol. 21, no. 2, pp. 365–375, 2012.
- [5] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328–348, 2009.
- [6] B. Akay and D. Karaboga, "A modified Artificial Bee Colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, no. 1, pp. 120–142, 2012.
- [7] M. Sonmez, "Artificial Bee Colony algorithm for optimization of truss structures," *Applied Soft Computing*, vol. 11, no. 2, pp. 2406–2418, 2011.
- [8] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5682–5687, 2010.
- [9] X. Liao, J. Zhou, S. Ouyang, R. Zhang, and Y. Zhang, "An adaptive chaotic artificial bee colony algorithm for short-term hydrothermal generation scheduling," *International Journal* of Electrical Power and Energy Systems, vol. 53, pp. 34–42, 2013.
- [10] S. Y. Liu, P. Zhang, and M. M. Zhu, "Artificial bee colony algorithm based on local search," *Control and Decision*, vol. 29, no. 1, pp. 123–128, 2014.
- [11] L. Bao and J. C. Zeng, "Comparison and analysis of the selection mechanism in the artificial bee colony algorithm," in *Proceedings of the Ninth International Conference on Hybrid Intelligent Systems*, pp. 411–416, Salamanca, Spain, June 2009.
- [12] H. Wang, W. Wang, and Z. Wu, "Particle swarm optimization with adaptive mutation for multimodal optimization," Applied Mathematics and Computation, vol. 221, pp. 296–305, 2013.
- [13] H. Wang, H. Li, Y. Liu, C. Li, and S. Zeng, "Opposition-based particle swarm algorithm with Cauchy mutation," in

- Proceedings of the IEEE Congress on Evolutionary Computation, pp. 4750–4756, Singapore, September 2007.
- [14] X. Y. Zhou, W. U. Zhi-Jian, H. Wang et al., "Elite opposition-based particle swarm optimization," *Acta Electronica Sinica*, vol. 41, no. 8, pp. 1647–1652, 2013.
- [15] A. Rajasekhar, A. Abraham, and M. Pant, "Levy mutated Artificial Bee Colony algorithm for global optimization," in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 655–662, Anchorage, AK, USA, October 2011.
- [16] M. S. Kıran and M. Gündüz, "A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems," Applied Soft Computing Journal, vol. 13, no. 4, pp. 2188–2203, 2013.
- [17] H. Duan, C. Xu, and Z. H. Xing, "A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems," *International Journal of Neural Systems*, vol. 20, no. 1, pp. 39–50, 2010.
- [18] S. M. Chen, A. Sarosh, and Y. F. Dong, "Simulated annealing based artificial bee colony algorithm for global numerical optimization," *Applied Mathematics and Computation*, vol. 219, no. 8, pp. 3575–3589, 2012.
- [19] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied mathematics and computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [20] F. Kang, J. Li, and Q. Xu, "Structural inverse analysis by hybrid simplex artificial bee colony algorithms," *Computers and Structures*, vol. 87, no. 13, pp. 861–870, 2009.
- [21] F. Kang, J. Li, and Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions," *Information Sciences*, vol. 181, no. 16, pp. 3508– 3531, 2011.
- [22] W. F. Gao, S. Y. Liu, and L. L. Huang, "A novel artificial bee colony algorithm with Powell's method," *Applied Soft Computing*, vol. 13, no. 9, pp. 3763–3775, 2013.

















Submit your manuscripts at www.hindawi.com























