Cluster Computing (2019) 22:S3971–S3980
https://doi.org/10.1007/s10586-018-2558-4

Improved artificial bee colony algorithm based on self-adaptive random optimization strategy

Wen Liu[1] · Tuqian Zhang[2,3] · Yan Liu[1] · Ningning Zhang[1] · Hongyu Tao[3] · Guoqing Fu[2,3]

Received: 18 January 2018 / Revised: 26 February 2018 / Accepted: 14 March 2018 / Published online: 23 March 2018

Abstract

In order to effectively overcome the disadvantages of the traditional artificial bee colony (ABC) algorithm, i.e., its tendency to fall into local optima and low search speed, an improved ABC algorithm based on the self-adaptive random optimization strategy (SRABC) is proposed. First, the improved algorithm was derived from the self-adaptive method to update the new location of an ABC to improve the correlation within the bee colony. It converges swiftly and obtains the optimal solution for the benchmark function. Second, the bidirectional random optimization mechanism was used to restrain the search direction for the fitness function in order to improve the local search ability. Moreover, the particle swarm optimization algorithm regarded as the initial value of the SRABC algorithm was introduced at the initial stage of the improved ABC algorithm to increase the convergence rate, search precision and searchability, and greatly reduce the search space. Finally, simulation results for benchmark functions show that the proposed algorithm has obviously better performance regarding the search ability and convergence rate, which also prevents early maturing of algorithm.

Keywords Swarm intelligence · Artificial bee colony (ABC) · Bidirectional random optimization (BRO) · Self-adaptive · Particle swarm optimization (PSO)

1 Introduction

Optimization problems have always been widely used in mathematical programming, computer technology, engineering design, etc. Moreover, the solution of an optimization problem with simulations of group behaviors has become a popular research topic. Among the possible simulation techniques, Ant Colony Optimization (ACO) [1] proposed by Dorgo and particle swarm optimization (PSO) [2] proposed by Kennedy are relatively mature in this field.

In recent years, a new random search method in the optimization field has been developed—namely, artificial bee colony (ABC) algorithm, which was proposed by Karaboga and successfully applied to the optimization problem of functions [3]. It mainly simulates a bee colony that gathers honey intelligently and exchanges information about nectar sources according to the different divisions of labor within the bee colony until the optimal nectar source is found. In references [4, 5], the performance of the ABC algorithm was tested with five common benchmark functions, showing that the ABC algorithm has good optimal performance similar to the Genetic Algorithm (GA), PSO,

✉ Wen Liu
627952@qq.com

Tuqian Zhang
245632154@qq.com

Yan Liu
719261357@qq.com

Ningning Zhang
157267828@qq.com

Hongyu Tao
3415679205@qq.com

Guoqing Fu
1392752559@qq.com

[1] Department of Electrical and Information Engineering, Xinjiang Institute of Engineering, Urumqi 830091, China

[2] School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

[3] School of Science and Technology, Xinjiang Agricultural University, Urumqi 830091, China

and Differential Evolution (DE). However, it has disadvantages such as a low convergence rate and tendency to "mature early." Targeting these problems, the "early maturing" phenomenon in the ABC algorithm was overcome by combining it with the DE algorithm and introducing an elimination rule during iteration [6]. Chaotic thought was introduced to the ABC algorithm in references [7, 8], and the characteristics of chaotic motion, such as the randomness and ergodicity, were used to improve the overall searchability of the algorithm.

In order to improve the convergence rate of the ABC algorithm and overcome its tendency to fall into local optima in a later period, an improved algorithm based on the self-adaptive random optimization strategy (SRABC) was proposed on the basis of references [6, 9, 10] to improve the local searchability of the algorithm by taking advantage of self-adaptive thought and the bidirectional random optimization mechanism. In addition, PSO was introduced at the initial stage of the improved algorithm to increase its convergence rate.

In order to improve the exploitation capability, the chaotic gradient ABC algorithm was proposed in Ref. [11], its effectiveness was tested on three different benchmark text datasets namely Reuters-21,578, Classic4, and WebKB. Astute Artificial Bee Colony (AsABC) algorithm proposed by Ref. [12] circumvents three problems: slow convergence speed, local optima stagnation and scalability problem. What's more it is able to maintain a better trade-off between two conflicting aspects, exploration and exploitation in the search space.

A decentralized form of ABC algorithm with dynamic multi-populations by means of fuzzy C-means (FCM) clustering proposed by Ref. [13] improves the convergence rate and makes a balance between the global search and local turning abilities. A novel ABC algorithm with dynamic population (ABC-DP) proposed by Ref. [14–16] synergizes the idea of extended life-cycle evolving model to balance the exploration and exploitation tradeoff. ABC-DP is then used for solving the optimal power flow (OPF) problem in power systems that considers the cost, loss, and emission impacts as the objective functions. Taking Kapur's entropy as the optimized objective function, Ref. [17–20] put forward the modified quick ABC algorithm (MQABC), which employed a new distance strategy for neighborhood searches.

In order to overcome the poor exploitation of ABC algorithm, inspired by the gravity model, ABCG was proposed on basis of a novel solution search equation and multiple solution search equations. The results of testing on benchmark functions show that ABCG is regarded as a competitive solution [21].

A novel strategy based on node electrical relevance and ABC algorithm was proposed to minimize outage losses and make use of renewable energy sources. The results of testing on the IEEE 69-bus distribution system show that the proposed strategy is more feasible and efficient than other strategies from the literature [9].

On basis of the ABC algorithm, a new topological shape optimization scheme was proposed. The algorithm is the most effective among the comparison methods [22].

In order to overcome the weakness of slow convergence in ABC, CosABC based the cosine similarity was proposed. Via testing on a test suite composed of twenty-four benchmark functions, the effectiveness of CosABC is demonstrated [7].

Recommender systems make an important role in electronic commerce sites, which helps to achieve better customer satisfaction and bring those products into the notice of the customer. A movie recommender system based on collaborative filtering technique was presented. On the basis of CPU Time and two standard functions, ant colony optimization and artificial bee colony optimization were compared.

The rest of paper is organized as follows. Section 1 presents the ABC algorithm, and Sect. 2 presents the essential concepts of the SRABC and the strategy for self-adaptive random optimization. Section 3 discusses the performance of the proposed approach for various benchmark functions. Finally, the conclusions are provided in Sect. 5.

2 Principles of the ABC algorithm

The ABC algorithm is a random search algorithm based on the behaviors of a bee colony [3]. In the ABC algorithm, an ABC falls into three categories according to the labor distribution: honey-gathering, observation, and investigation bees. Generally, both the honey-gathering and observation bees account for half of the bee colony, and there is only one honey-gathering bee for one nectar source. During the initialization of the algorithm, M initial solutions are first generated randomly; then, the optimal solution is found with the following search process: (1) the honey-gather bee searches and memorizes the amount of honey in the nectar source—namely, the quality of the solution (fitness); (2) the observation bee selects a nectar source by determining the yield rate according to the information about the nectar source obtained by the honey-gathering bee and changes the position memorized; and (3) when a nectar source is given up after it is exhausted, an investigation bee is produced to search for new nectar source.

In the algorithm, in order to generate a new candidate position V_i according to the memorized position X_i, the following equation is used for updating:

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

where k is a nectar source different from i, j is the subscript of a randomly selected nectar source, and φ_{ij} is a random number in the range [− 1, 1]. According to the amount of honey at the nectar source, the probability at which a nectar source is selected by the observation bee is

$$p_i = \frac{fit(\theta_i)}{\sum_{i=1}^{S} fit(\theta_i)} \tag{2}$$

where S is the total number of nectar sources, θ_i is the ith nectar source, $f(\theta_i)$ is the fitness of nectar source θ_i, and $i \in$ P{1, 2, …, S}.

Suppose that after “limit” times of cycling search and update, the fitness of the nectar source could still not be improved; then, it would be given up, and the honey-gathering bee would turn into an investigation bee. “Limit” is an important control parameter in the ABC algorithm for the selection of the investigation bee. The procedure for the investigation bee finding a new nectar source and replacing X_i is expressed as follows:

$$x_i^j = x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j) \tag{3}$$

3 Improvement in the ABC algorithm

In order to improve the convergence rate of the ABC algorithm and overcome its tendency to fall into local optima in a later period, an improved algorithm based on the self-adaptive random optimization strategy is proposed on the basis of the above-stated ABC algorithm principle. Together with weighted PSO, the convergence rate and local searchability of the algorithm are improved by taking advantage of self-adaptive thought and bidirectional random optimization to improve the optimal search performance of this algorithm.

3.1 PSO algorithm

In 1995, after the proposal of the basic PSO algorithm by Zhang et al. [9], an improved PSO algorithm was proposed by Shi and Eberhart [10]; in it, an inertia factor w is added to the basic PSO algorithm, and the updated equations for standard PSO are expressed as

$$V_i^{t+1} = wV_i^t + c_1 r_1 (P_i^t - X_i^t) + c_2 r_2 (P_g^t - X_i^t) \tag{4}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{5}$$

where V_i is the speed of particle i, X_i is the position of particle i, c_1 and c_2 are positive acceleration factors, r_1 and r_2 are random numbers distributed within [0, 1], P_i is the optimal position found by a single particle, P_g is the optimal position found by the entire colony, t is the *tth* iteration. The basic steps of the specific optimization process for the PSO algorithm [22] are as follows:

Step 1: Initialize PSO, set the initial position and speed of m particles randomly, and calculate the fitness of each particle.

Step 2: For each particle, compare the fitness of the current position with the fitness at the best position P_i—namely, $P_{i,best}$. If it is better than $P_{i,best}$, update P_i and $P_{i,best}$; otherwise, $P_{i,best}$ remains unchanged.

Step 3: For each particle, compare the fitness of the current position with the fitness at the best position P_g—namely, $P_{g,best}$. If it is better than $P_{g,best}$, update P_g and $P_{g,best}$; otherwise, $P_{g,best}$ remains unchanged.

Step 4: The speed and position of the particle are adjusted according to Eqs. (4) and (5).

Step 5: If the fitness of the current position reaches the termination condition, it would end or return to Step 2.

3.2 Self-adaptive random optimization strategy

3.2.1 Position-update equation for a self-adaptive bee colony

When adopting Eq. (1) for a position update, a larger φ_{ij} causes movement away from the local minimum, whereas a smaller φ_{ij} favors the convergence of the algorithm [22]. The best method for the overall search is to adopt a larger φ_{ij} at the initial stage of the algorithm to obtain an excellent nectar source with a higher searchability and improve the search precision. In a later period, a smaller φ_{ij} is needed to improve the local searchability of the algorithm and increase its convergence rate. Therefore, φ_{ij} is set as the function for iteration, and it decreases as the number of iterations increases. φ_{ij} is defined as follows:

$$\varphi_{ij}^k = \varphi_{ij}^{k-1} - \frac{C(w_{\max} - w_{\min})}{C_{\max}} \tag{6}$$

where $w_{\max}$ and $w_{\min}$ are initial and final weights, respectively; $C_{\max}$ is the maximum number of iterations; and C is the current number of iterations. As a result, Eq. (1) is redefined as.

$$v_{ij} = x_{ij} + \varphi_{ij}^k (x_{ij} - x_{kj}) \tag{7}$$

Thus, to some degree, Eq. (7) plays a guiding role in the search trend for the position of a nectar source, overcoming disadvantages such as the strong randomness and low convergence rate.

3.2.2 Bidirectional random optimization mechanism

When calculating the fitness of a nectar source with the ABC algorithm, the observation bee selects a nectar source after comparing the ones around θ_i. The position near the nectar source is calculated as follows:

$$\theta_i(C+1) = \theta_i(C) + \varphi_i(C) \tag{8}$$

where $\varphi_i(c)$ is the progressive step length produced randomly near θ_i. After calculating the fitness, if *fit*(C + 1) $>$ *fit*(C), then the observation bee would choose θ_i(C + 1), or $\varphi_i(c)$ remains unchanged. If the fitness of a nectar source is not improved after cycling for a finite number of iterations, then it should be given up, and the honey-gathering bee would turn into an investigation bee according to Eq. (3). There are certain disadvantages associated with the above-stated method; namely, in each cycle, the nectar source in a single direction would be searched, and as a result, there is a tendency to fall into local optima. In references [23, 24], a bidirectional random optimization mechanism was proposed in research on the search hit rate and success rate in a dynamic network environment [23], which effectively improved the search feature of the network. Inspired by this thought, an improved mechanism is introduced to improve the search direction for a nectar source; if

$$fit(\theta_{\mathrm{i}} + l) < fit(\theta_{\mathrm{i}}) \tag{9}$$

then

$$\theta_{\mathrm{i}} = \theta_{\mathrm{i}} + d \tag{10}$$

If

$$fit(\theta_{\mathrm{i}} - l) < fit(\theta_{\mathrm{i}}) \tag{11}$$

then

$$\theta_{\mathrm{i}} = \theta_{\mathrm{i}} - d \tag{12}$$

or θ_i remains unchanged.

3.2.3 Algorithm initialization realized by the particle swarm optimization algorithm

The convergence rate is low in the bee colony algorithm, whereas it is relatively higher in the PSO algorithm, which is introduced in the initial stage to improve the algorithm. That is, the overall optimal solution is obtained by iteration by taking advantage of PSO; then, the position of a nectar source would be randomly generated near the optimal solution. Later, an optimization process is conducted to calculate the position of a nectar source within an ABC. The improved initial position of the nectar source with the ABC algorithm is as follows:

$$X_i = P^M_{g,best} + \varphi^M_i \cdot P_{g,best} \tag{13}$$

where $P^M_{g,best}$ is an M-dimensional vector, and each element is equal to $P_{g,best}$. φ^M_i is an M-dimensional vector in [− 1, 1] produced randomly. According to the above-mentioned strategy, the specific procedures for improving the algorithm are as follows:

Step 1 The related initial parameters for the ABC and PSO algorithm are set. The initial speeds and positions of M particles are randomly generated according to Eqs. (4) and (5).

Step 2 The optimal solution $P_{g,best}$ within the number of cycles c is determined by calculating the fitness value of each particle for comparison.

Step 3 The honey-gathering bee searches for a new nectar source according to Eqs. (6) and (7) and calculates its fitness. If it is better than the original position, then the original position is replaced with the new one.

Step 4 The observation bee selects a nectar position according to the amount of honey in the nectar source according to the probability in Eq. (2), generates a new position according to the bilateral random optimization mechanism, and evaluates this position.

Step 5 If the nectar source is given up, then the honey-gathering bee at this nectar source would turn into an investigation bee according to Eq. (3).

Step 6 The current optimal position and fitness value are recorded.

4 Analysis of the simulation experiment

In order to verify the effectiveness of the proposed algorithm and conduct a performance analysis, seven benchmark functions are selected for comparison and testing, which are different from the traditional ABC algorithm. A comparison of the performance of a hybrid ABC is proposed in reference [25].

(1) Rastrigin function $f(x) = \left(\sum_{i=1}^{D} x_i^2 - 10\cos(2\pi x_i) + 10\right)$ is a multimodal function whose optimal solutions are distributed evenly with the search scope within [− 20, 20] and the overall optimal solution of 0.

(2) Griewank function $f(x) = \frac{1}{4000}\left(\sum_{i=1}^{D} x_i^2\right) - \left(\prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}})\right) + 1$ is a multimodal function whose optimal solutions are distributed evenly. However, its local optima increase as the number of dimensions increases.

The search scope is within [− 600, 600], and the optimal solution is 0.

(3) Sphere function

$f(x) = \sum_{i=1}^{n} x_i^2$ is a continuous convex function with a single peak whose search scope is within [− 100, 100]. The lowest point of the function is 0.

(4) Rosenbrock function

$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2];$ the search scope is within [− 30, 30], and the optimal solution is 0.

(5) Ackley function

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

; the search scope is within [− 30, 30], and the lowest point of the function is 0.

(6) Pathological function

$$f(x) = \sum_{i=1}^{n-1}\left(0.5 + \frac{\sin^2(\sqrt{100x_i^2 + x_{i+1}^2}) - 0.5}{(1 + 0.001(x_i^2 + x_{i+1}^2 + 2x_i x_{i+1})^2)}\right)$$

; the search scope is within [− 100, 100], and the lowest point of the function is 0.

(7) Alpine function

$f(x) = \sum_{i=1}^{D} |x_i \sin(x_i) + 0.1x_i|$; the search scope is within [− 10, 10], and the lowest point of the function is 0.

The parameters for the algorithms are set as follows. For the SRABC algorithm, the scale of the colony S = 60, limit = 60, the number of PSOs is 60, and the number of cycles for the PSO algorithm c = 100. For the standard ABC algorithm S = 60, limit = 60, the maximum number of iterations is 2500, and the algorithm operates 30 times independently. A comparison of the maximum, minimum, average, and variance obtained after the independent operation of various functions in different dimensions 30 times is summarized in Table 1.

From Table 1, although the optimization results for the Sphere function with a single peak for different dimensions do not greatly improve with the SRABC algorithm, it is still better than those obtained by the standard ABC and the hybrid artificial bee colony (HABC) algorithms. For the multimodal Griewank and Rastrigin functions, there are complicated nonlinear overall optimization problems. From the table, the precision of the simulation results of seven functions for different dimensions with the SRABC algorithm is better than those of the ABC and HABC algorithms. In particular, the 60-dimensional Rastrigin function converges to 0 rapidly. The overall optimal solution of the Rosenbrock function is distributed in a long, narrow, and flat parabolic valley, and it is difficult to converge to this solution. From Table 1, although the average values of the Ackley function, f the Alpine function and the Pathological function are large, the average values of the proposed algorithm are obviously smaller than the other two, and the minimum is close to the optimal solution. From the test of the Ackley function, the performance of the SRABC algorithm is not much better than that of ABC and HABC algorithms when the number of dimensions is low; however, as the number of dimensions increases, the precision of the optimal solution of the SRABC algorithm is obviously better than the other two algorithms. From the test of the Alpine function in Table 1, the performance of the HABC algorithm is better than that of the ABC algorithm, and the performance of the proposed SRABC algorithm is much better than the other two. From the test of the Pathological function in Table 1, the search precision of the proposed algorithm is more stable than the other two when the number of dimensions is 30, and there is not much difference between the three algorithms when the number of dimensions is 60. Further, from Table 1, the improved algorithm maintains the features of its original algorithm and improves the calculation precision and stability compared to the traditional and HABC algorithms.

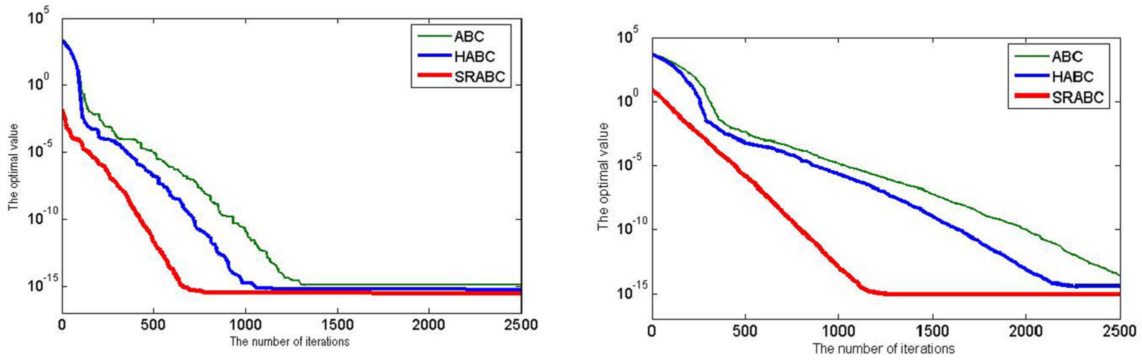
In Figs. 1, 2, 3, 4, 5, 6 and 7, the curves showing the optimal values obtained by the ABC, HABC, and SRABC algorithms versus the number of iterations for each function with 30 and 60 dimensions are shown. In order to show the results clearly, the number of iterations is plotted along the x axis, and the optimal values are plotted along the y axis on a logarithmic scale.

In Figs. 1, 2, 3, 4, 5, 6 and 7, for the same number of iterations and targeting different dimensions of seven benchmark functions, iteration of the optimal value nearly stops in a later period, and there is a certain improvement obtained by the HABC algorithm compared to the ABC algorithm. Further, the performance of the SRABC algorithm is greatly improved by the proposed optimization process. From Fig. 1, the SRABC algorithm converges swiftly and obtains the optimal solution for the Sphere function. Since the optimization result at the initial stage of the PSO algorithm is regarded as the initial value of the SRABC algorithm, the search space is greatly reduced, and the convergence rate is further improved. From Figs. 2 and 3, for a certain number of iterations, the optimal values could be achieved by the SRABC algorithm at an early

Table 1 Test results for different functions

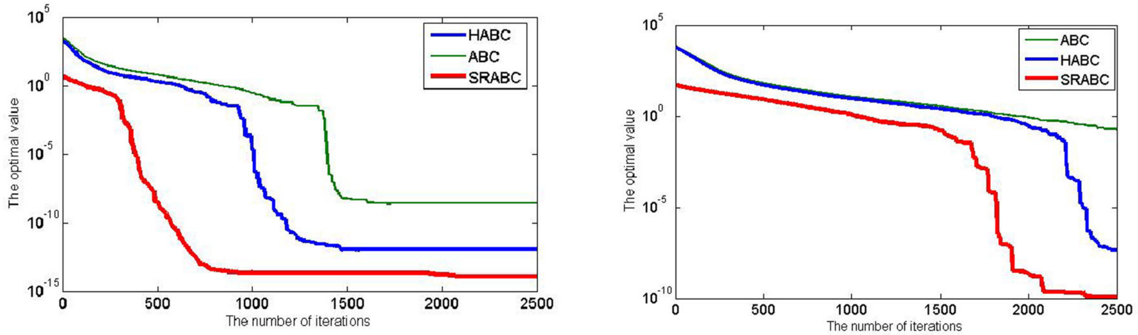
Test function	Algorithm	Dimension	Average value	Variance	Maximum value	Minimum value
(f_1) Sphere function	ABC	30	1.13714E−015	2.97144E−016	1.79207E−015	5.46909E−016
	HABC		5.55572E−016	9.00306E017	7.23074E−016	4.09377E−016
	SRABC		2.95155E−016	5.79213E−17	4.61564E−016	1.80046E−016
	ABC	60	2.34037E−014	1.82907E−014	7.64184E−014	4.13668E−015
	HABC		3.6088E−015	1.45665E−015	8.65843E−015	1.87149E−015
	SRABC		8.15309E−016	1.32347E−016	9.89696E−016	4.72298E−016
(f_2) Rastrigin function	ABC	30	2.58239E−009	1.03788E−008	5.65922E − 008	1.13687E−012
	HABC		1.13687E−012	1.64169E−012	8.15703E−012	1.12578E−015
	SRABC		1.32635E−014	1.78719E−14	5.68434E−014	0
	ABC	60	0.167895	0.458241	1.99033	1.72463E−010
	HABC		4.56788E−008	1.53871E−007	8.25186E−007	1.3074E−011
	SRABC		1.21393E−010	2.62695E−010	1.2559E−009	1.7053E−013
(f_3) Griewank function	ABC	30	2.18149E−012	9.27583E−012	4.90501E−011	9.99201E−016
	HABC		4.08007E−014	9.89653E−014	4.99267E−013	4.44089E−016
	SRABC		6,25426E016	1.41953E−015	7.54952E−015	0
	ABC	60	2.79554E−013	8.89572E−013	4.85578E−012	2.77556E−015
	HABC		1.4011E−013	2.87342E−013	1.20581E−012	2.33147E−015
	SRABC		1.72825E−015	1.94662E−015	8.54872E−015	1.11022E−016
(f_4) Rosenbrock function	ABC	30	0.250332	0.235661	1.04043	0.0199202
	HABC		0.247764	0.277462	1.43555	0.0262511
	SRABC		0.0142749	0.0138214	0.0513423	0.0013952
	ABC	60	4.3507	3.10707	11.5256	0.344054
	HABC		1.37106	1.12131	3.99565	0.114811
	SRABC		0.0300207	0.0329917	0.140948	0.00193665
(f_5) Ackley function	ABC	30	1.00483E−013	2.62522E−014	1.83853E−013	6.30607E−014
	HABC		6.30607E−014	1.07237E−014	7.72715E−014	4.88498E−014
	SRABC		5.50671E−015	1.49796E−015	6.21725E−015	2.66454E−015
	ABC	60	2.88426E−007	1.1748E−007	5.20881E−007	1.55876E−007
	HABC		3.74634E−09	1.11671E−010	5.88862E−010	2.44249E−09
	SRABC		1.00625E−012	1.41953E−013	1.00755E−012	1.053214E−013
(f_6) Alpine function	ABC	30	5.09234E−−007	7.75053E−007	2.52902E−006	1.18207E−008
	HABC		1.5524E−010	6.28861E−010	3.45098E−009	1.85747E−015
	SRABC		3.63054E−014	6.75367E−014	2.48683E−013	5.55035E−016
	ABC	60	0.0248383	0.0254136	0.0939726	0.000341641
	HABC		0.00120275	0.00241429	1.58433E−006	0.0123575
	SRABC		8.13672E−07	8.46449E−07	6.36929E−07	3.35827E−04
(f_7) Athological function	ABC	30	0.00817639	0.00873776	0.029539	2.23771E−005
	HABC		0.00237032	0.00266042	0.00884325	6.21676E−005
	SRABC		1.11085E−005	1.13333E−005	3.31772E−005	5.36175E−008
	ABC	60	0.103806	0.0385138	0.210675	0.085155
	HABC		0.0790134	0.0195857	0.0958031	0.036461
	SRABC		0.0164499	0.0185505	0.0745268	6.4964E−005

stage for the Griewank and Rastrigin functions, which nearly decrease linearly, and the convergence rate is smaller than those of the other two algorithms. From Figs. 4 and 5, the SRABC algorithm has a greater precision than the ABC and HABC algorithms for 30 dimensions if the number of iterations is low. Moreover, as the number of iterations increases, the SRABC algorithm converges to the optimal value gradually and steadily. The proposed



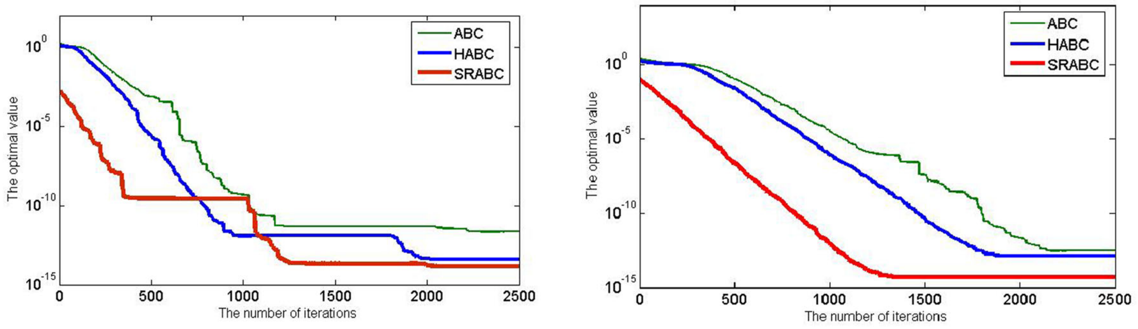
(a) D = 30 **(b)** D = 60

Fig. 1 Comparison of different dimensions for the Sphere function



(a) D = 30 **(b)** D = 60

Fig. 2 Comparison of different dimensions for the Rastrigin function



(a) D = 30 **(b)** D = 60

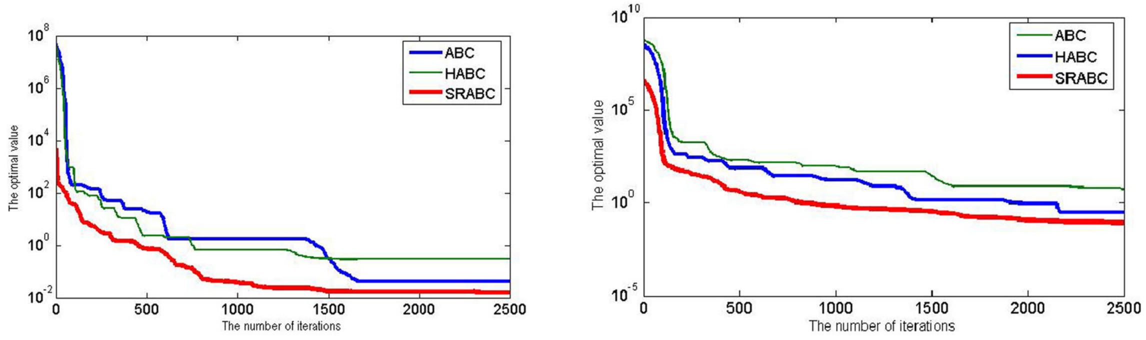
Fig. 3 Comparison of different dimensions for the Griewank function

algorithm could perform more steady astringency and local or overall searchability when the number of dimensions is 60. From Figs. 6 and 7, compared with the other two algorithms, the proposed algorithm ensures an initial search precision and converges to the optimal value as the number of iterations increases.

The self-adaptive random optimization strategy for a bee colony could increase the convergence to an optimal value during the entire optimization process, which could guide individuals to the overall optimal value. There are substantial improvements in the precision and convergence rate when comparing the ABC and SRABC algorithms, which also prevents early maturing of algorithm.

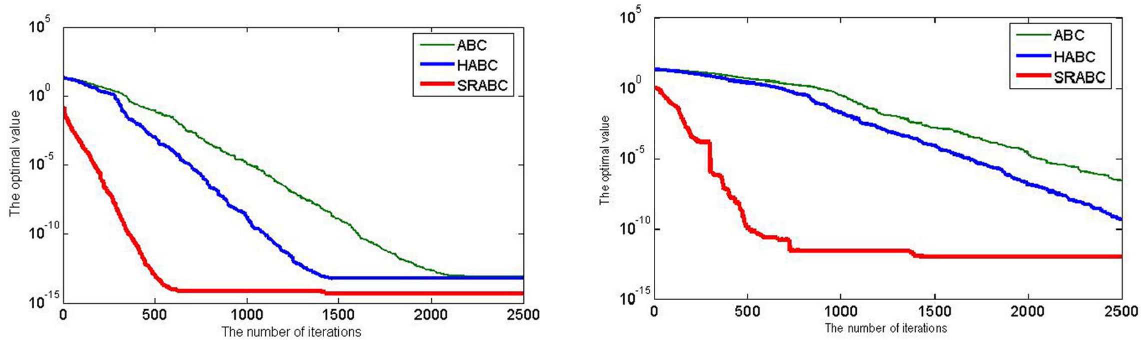
5 Conclusion

As a novel intelligent swarm optimization algorithm, the ABC algorithm is characterized by its easy realization, simple operation, and few control parameters [21, 26].



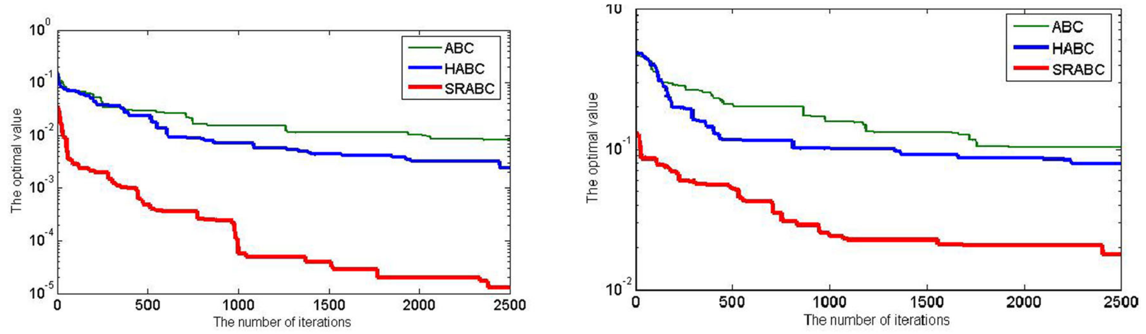
(a) D = 30 **(b)** D = 60

Fig. 4 Comparison of different dimensions for the Rosenbrock function



(a) D = 30 **(b)** D = 60

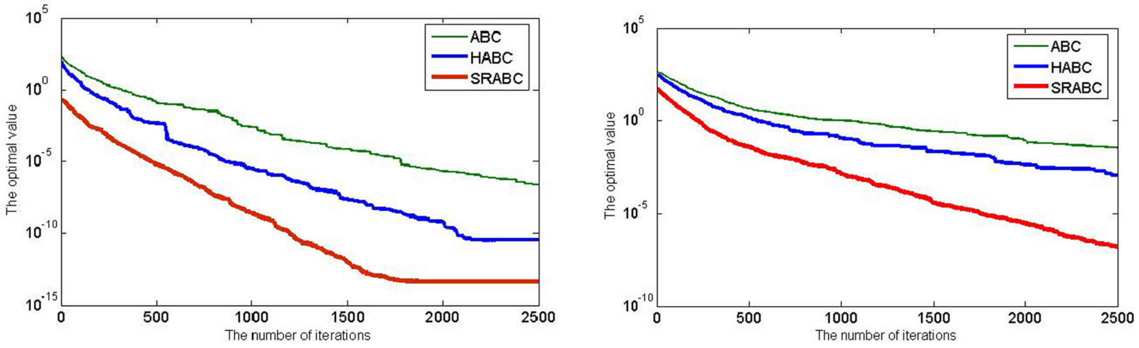
Fig. 5 Comparison of different dimensions for the Ackley function



(a) D = 30 **(b)** D = 60

Fig. 6 Comparison of different dimensions for the Alpine function

Targeting the weak local searchability, low search precision, and low convergence rate of the ABC algorithm, a PSO algorithm was introduced at the initial stage to initialize the bee colony. On the basis of self-adaptive thought and the bidirectional random optimization mechanism, an improved algorithm based on the bidirectional random optimization strategy was proposed to effectively overcome disadvantages such as the strong randomness and single search direction during the optimization process, which avoided the tendency to fall into local optima to a certain degree. Through a comparison with the traditional algorithm and the HABC algorithm proposed in Ref. [6], the proposed algorithm was generally found to be effective for seven different benchmark functions, which improved optimization ability of algorithm on the basis of raising convergence rate of algorithm.



(a) D = 30

(b) D = 60

Fig. 7 Comparison of different dimensions for the Pathological function

Acknowledgements Part of the results in this paper appeared in the Proceedings of the 9th International Symposium on Computational Intelligence and Design (ISCID), 2016. This work is supported by the Scientific Research Program of the Higher Education Institution of Xinjiang under Grant No. XJEDU2016I049, the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant No. 2017D01B09, Youth Research start-up fund project of School of Science and Technology Xinjiang Agricultural University under Grant No. 2016KJKY006 and No. 2016KJKY007.

References

1. Dorgo, M., Maniezzo, V., Colorni, A.: The ants system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B **26**(1), 29–41 (1996)
2. Kennedy, J., Ebethart, R.: Particle swarm optimization. In: Proceeding of IEEE International Conference on Neural Networks. IEEE Computer Society, Piscataway, pp. 1942–1948 (1995)
3. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, Technical Report-TR06. Computer Engineering Department, Engineering Faculty, Erciyes University (2005)
4. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial Bee Colony(ABC) algorithm. J. Global Optim. **39**(3), 459–471 (2007)
5. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. **8**(1), 687–697 (2008)
6. Abraham, A., Jatoth, R.K., Rajasekhar, A.: Hybrid differential artificial bee colony algorithm. J. Comput. Theor. Nanosci. **9**(2), 1–9 (2013)
7. Roeva, O.: Application of Artificial Bee Colony Algorithm for Model Parameter Identification. Springer, Cham (2018)
8. Xu, C.F., Dun, H.B., Liu, F.: Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning. Aerosp. Sci. Technol. **14**(8), 535–541 (2010)
9. Zhang, H., Peng, M., Wu, H., et al.: A strategy for intentional islanding of distribution networks based on node electrical relevance and artificial bee colony algorithm. IEEJ Trans. Electr. Electron. Eng. **13**(2), 84–91 (2018)
10. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proceedings of IEEE World Congress on Evolutionary Computation, Anchorage, Alaska, USA, pp. 69–73, May 1998
11. Bharti, K.K., Singh, P.K.: Chaotic gradient artificial bee colony for text clustering. Soft. Comput. **20**(3), 1113–1126 (2016)
12. Kishor, A., Chandra, M., Singh, P.K.: An astute artificial bee colony algorithm. J. Adv. Intell. Syst. Comput. **546**, 153–162 (2017)
13. Zhang, M., Ji, Z., Wang, Y.: Artificial bee colony algorithm with dynamic multi-population. Mod. Phys. Lett. B **31**, 1740087 (2017)
14. Ding, M., Chen, H., Lin, N., et al.: Dynamic population artificial bee colony algorithm for multi-objective optimal power flow. Saudi J. Biol. Sci. **24**(3), 703 (2017)
15. Huo, F., Liu, Y., Wang, D., et al.: Bloch quantum artificial bee colony algorithm and its application in image threshold segmentation. Signal Image Video Process. **11**(12), 1–8 (2017)
16. Kefayat, M., Ara, A.L., Niaki, S.A.N.: A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources. Energy Convers. Manag. **92**(3), 149–161 (2015)
17. Ozturk, C., Hancer, E., Karaboga, D.: Dynamic clustering with improved binary artificial bee colony algorithm. Appl. Soft Comput. **28**, 69–80 (2015)
18. Gao, K.Z., Suganthan, P.N., Pan, Q.K., et al.: An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time. Expert Syst. Appl. **65**(C), 52–67 (2016)
19. Li, J.Q., Pan, Q.K., Duan, P.Y.: An improved artificial bee colony algorithm for solving hybrid flexible flow shop with dynamic operation skipping. IEEE Trans. Cybern. **46**(6), 1311–1324 (2016)
20. Zhang, P., Li, J., Hu, X., et al.: Crossover-based artificial bee colony algorithm for constrained optimization problems. Neural Comput. Appl. **26**(7), 1587–1601 (2017)
21. Xiang, W.L., Meng, X.L., Li, Y.Z., et al.: An improved artificial bee colony algorithm based on the gravity model. Inf. Sci. **429**, 49–71 (2018)
22. Kim, Y.H., Han, S.Y.: Topological shape optimization scheme based on the artificial bee colony algorithm. Int. J. Precis. Eng. Manuf. **18**(10), 1393–1401 (2017)
23. Wen Ming, M.A., Meng, X.W., Zhang, Y.J.: Bidirectional random walk search mechanism for unstructured P2P network. J. Softw. **23**(4), 894–911 (2013)
24. Jia, Z., Si, X., Wang, T.: Optimum method for sea clutter parameter based on artificial bee colony. J. Cent. South Univ. (Sci. Technol.) **43**(9), 3485–3489 (2012)
25. Sethi, D., Singhal, A.: Comparative analysis of a recommender system based on ant colony optimization and artificial bee colony optimization algorithms. In: International Conference on Computing, Communication and Networking Technologies, pp. 1–4. IEEE Computer Society (2017)

26. Zhao, Z., Huang, W.: Improved artificial bee swarm algorithm and its application in optimal operation of wind-power generators. J. Cent. South Univ. (Sci. Technol.) **42**(10), 3101–3104 (2011)

Wen Liu received the B.S. degree in computer science from Xinjiang Normal University, Xinjiang, in 2004 and the Ph.D. degree in computer science from Dalian University of Technology, Dalian, in 2017. He is currently an associate Professor in Department of Electrical and Information Engineering, Xinjiang Institute of Engineering. His research interests include swarm intelligence, stream data processing, and cloud computing.

Tuqian Zhang received the B.S. degree in Electronic Information science from West Anhui University, Anhui, in 2008 and the M.S. degree in Information System from Xinjiang Agricultural University, Xinjiang, in 2012. He is currently a second M.S. degree student of computer science at Dalian University of Technology. His research interests include electronic communication, database and cloud computing.

Yan Liu received the B.S. degree in Electrical engineering and automation from Shihezi University, Xinjiang, in 2005 and the Master's degree in control engineering from Xinjiang University, Xinjiang, in 2011. She is a lecturer currently in Department of Electrical and Information Engineering, Xinjiang Institute of Engineering. His research interests include automation technology, automatic detection, measurement and control instrument.

Ning-Ning Zhang received the B.S. degree in automation from Xinjiang University, Xinjiang, in 2006 and the M.S. degree in control theory and control engineering from Xinjiang University, xinjiang, in 2009. She is currently an lecturer in Department of Electrical and Information Engineering, Xinjiang Institute of Engineering. Her research interests include data processing, and intelligent control.

Hongyu Tao received the B.S. degree in Information management and information systems from Xinjiang Agricultural University, Xinjiang, in 2016. Her research interests include database and cloud computing.

Guoqing Fu received the B.S. degree in computer science and technology from Xinjiang Finance and Economics University, Xinjiang, in 2014. He is currently the M.S. degree student of electronic and communication engineering at Dalian University of Technology. His research interests include electronic communication and cloud computing.