ISSN: 2664-9640

An Improved Artificial Bee Colony Algorithm Based on Elite Solution and Random Individual Neighborhood Information

Bingjie Wei [a], Qingtao Wu [b]

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China;

[a]bjwei@stu.haust.edu.cn, [b]wqt8921@haust.edu.cn

Abstract

Due to some shortcomings of the artificial bee colony algorithm(ABC), such as poor development ability and slow convergence speed, this article gives useful information on the use of the elite solution of randomly selected individuals and their neighborhoods in the bee phase, and the use of the group in the bystander bee phase. The search equation for the information of the optimal solution. The proposed search equations not only accelerate the convergence speed of the improved algorithm to some extent, but also guarantee the exploration ability of the algorithm in a certain sense due to the introduction of randomly selected individuals. The simulation results of 22 benchmark functions demonstrate that the proposed algorithm is superior to the comparison algorithm on most test functions.

Keywords

Artificial bee colony algorithm, Euclidean distance, elite solution, neighborhood information.

1. Introduction

With the advancement of science and technology, the optimization problem becomes more and more complicated. The optimization problem is characterized by non-convexity, discontinuity, non-differentiability and multi-mode [1]. Therefore, it is a great challenge to solve these global optimization problems using traditional derivative-based methods. Evolutionary algorithm as a new derivative-free optimization technology is proposed to solve these complex problems. In the field of evolutionary algorithms, some mainstream algorithms include Differential Evolution Algorithm (DE) [2], Genetic Algorithm (GA) [3], Artificial Bee Colony Algorithm (ABC) [4], Particle Swarm Optimization (PSO) [5], etc.

ABC algorithm is a stochastic optimization method based on swarm intelligence proposed by D. Araboga [4] in 2005. Its inspiration comes from the swing dance of bees and intelligent foraging behavior, which has been proved to be a very effective method to solve complex optimization problems. In recent years, ABC algorithm has been successfully extended to solve many application problems such as multi-objective optimization [6], binary optimization [7] and data clustering [8] due to its advantages of outstanding performance, convenient implementation and simple structure. However, the convergence speed of ABC algorithm is slow, because the use of search equation is mainly for exploration, rarely for development. The balance between exploration and development can ensure the performance of an intelligent algorithm. Therefore, in order to make up for the deficiency of ABC algorithm and further improve the performance of ABC algorithm, an improved algorithm of ABC is proposed and a new search equation is given in [9,10]. Experiments show that these improved ABC algorithms do improve the performance of ABC algorithm. However, there is no single effective solution to every problem. Therefore, the performance of ABC algorithm still has a lot of room for improvement.

ISSN: 2664-9640

Aiming at the shortcoming of ABC algorithm, this paper proposes an improved artificial bee colony algorithm based on elite solution and random individual neighborhood information (ERABC) to further improve the performance of ABC algorithm. Generally speaking, the employed bee stage of ABC algorithm is more inclined to exploration, while the onlooker bee stage is more inclined to development. The two search equations in this paper also maintain this principle. In the employed bee stage, the optimal solution of the individual and its neighborhood is selected randomly to guide the bee's search. In the onlooker bee stage, the information of the elite solution, the current optimal solution and the neighborhood information of randomly selected individuals are used to guide the bee's search. In this way, not only the convergence speed of the algorithm is accelerated, but also the exploration ability of the algorithm is ensured to some extent by the introduction of random selection of individuals. In addition, in the stage of bee observation, we dispense with the step of roulette, and directly allocate the computational resources to high-quality food sources (elite solution), which saves the computational amount to a certain extent.

2. Basic Artificial Bee Colony Algorithm

ABC algorithm can be divided into four stages, and the details of each stage are as follows:

1)In the initialization stage, initialize the basic parameters(i.e., population number SN, dimension number D, maximum function evaluation times $\max.FES$, predetermined period $limit$). Then a set of food source locations which represent the possible solutions of the optimization problem is generated, as shown below:

$$x_{i,j} = x_j^{\min} + \text{rand}(0,1) \cdot \left(x_j^{\max} - x_j^{\min} \right), \tag{1}$$

where $i \in \{1,2,\cdots,SN\}$, $j \in \{1,2,\cdots,D\}$, $x_j^{\max}$ and $x_j^{\min}$ are the upper bound and lower bound of the j th-dimensional variable of the solution respectively.

2)In the employed bee stage, each employed bee occupies a location of a food source, and candidate food source locations (new solutions) are generated by employed bees according to the following search equation:

$$v_{i,j} = x_{i,j} + \varphi_{i,j} \cdot \left(x_{i,j} - x_{k,j} \right), \tag{2}$$

where $i \in \{1,2,\cdots,SN\}$, $j \in \{1,2,\cdots,D\}$, and $k \neq i$; $\varphi_{i,j}$ is a random number generated by set [-1,1].

3)In the onlooker stage, each onlooker bee selects a food source location for development by means of roulette according to the probability value of the location of the food source. The probability calculation formula is as follows:

$$P_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j}, \tag{3}$$

where $fitness_i$ is the fitness value of the i th food source. Theoretically, the greater the fitness value, the greater the selection probability.

We consider the minimization problem, and the fitness value of the food source is defined as follows:

$$fitness_i = \begin{cases} \dfrac{1}{1+f_i} & \text{if } f_i \geq 0; \\ 1+|f_i| & \text{otherwise,} \end{cases} \tag{4}$$

ISSN: 2664-9640

where $fitness_i$ and f_i are fitness value and objective function value of food source x_i respectively. Once the food source is selected, the location update operation is carried out according to Eq. (2).

4)In the scout bee stage, once the location (solution) of a food source cannot be updated within a predetermined period, then the food source will be discarded and a new food source location will be generated randomly according to Eq. (1).

3. Improved Artificial Bee Colony Algorithm ERABC

3.1. Study Motivation

The research motivation of this paper mainly comes from the literature [11]. In the ABC algorithm, each generation of bees is searching independently. In nature, an employed bee will use a waddle dance to notify its neighbors that it has found a high-quality food source. Therefore, in the hiring bee stage, the search equation can use some other useful information to guide the search, so as to accelerate the convergence speed and avoid falling into the local optimum at the same time. In this paper, a new search equation is designed, which uses the useful information of the elite solution and the information of the randomly selected individuals and the optimal solution of their neighborhood. In the ABC algorithm, the observer bees can fly to any food source for further exploration. However, in the bee observation stage, this paper dispense with the process of roulette and directly search the elite solution to reduce the selection pressure and running time. Hence, the search of onlooker bees uses global information of the whole population to realize. To sum up, this paper improves the optimization performance of ABC algorithm by changing the search equations of employed bees and onlooker bees.

3.2. Search Equation for Employed Bees

In the ABC algorithm, employed bees blindly search around their food source. To accelerate the convergence speed under the condition of population diversity, we rank the objective function values from small to large and take the food source corresponding to the top ranking T $(T = p \cdot SN,\ p \in (0,1))$ as the elite solutions. In the employed bee stage, the beneficial information of the elite solution and the randomly selected individual k and its neighborhood are used. The role of the elite solution is to accelerate convergence, and the introduction of k can strengthen exploration at the same time avoid falling into local optimum. The new employed bee search equation is shown as follows:

$$v_{i,j} = \frac{x_{knbest,j} + x_{e,j}}{2} + \varphi_{i,j} \cdot (x_{knbest,j} - x_{k,j}), \tag{5}$$

where x_e is a randomly selected elitist solution and $e \neq i$; x_k is randomly selected from the population and k satisfies $k \in \{1, 2, ..., SN\}, k \neq e \neq i$; $x_{knbest,j}$ is the optimal food source location with the smallest objective function value within the neighborhood range of the k -th individual; $j \in \{1, 2, ..., D\}$ is a randomly chosen subscript; $\varphi_{i,j}$ is a random number between [-1,1]. If x_{knbest} is empty, the search is performed using Eq. (2) of the ABC algorithm.

3.3. Search Equation for Onlooker Bees

In the stage of onlooker bee, each onlooker bee determines the location of a food source and develops it, which mainly depends on the probability value of the location of the food source and is realized by roulette. Although the roulette method has a high probability of choosing high-quality food sources, there is still a certain probability of choosing poor quality food sources. Therefore, it wastes a certain number of evaluation times and the convergence speed

ISSN: 2664-9640

is slow. Therefore, we omitted the roulette wheel in the onlooker bee stage and directly updated the elite solution. The new onlooker bee search equation is as follows:

$$v_{i,j} = \frac{x_{best,j} + x_{e,j}}{2} + \varphi_{i,j} \cdot (x_{knbest,j} - x_{k,j}), \tag{6}$$

where x_e is a randomly selected elitist solution and $e \neq k$; x_{best} is the contemporary optimal solution; $\varphi_{e,j}$ is a random number between [-1,1]. If x_{knbest} is empty, the search is performed using Eq. (2) of the ABC algorithm.

Noticed that Eq. (6) is very similar to the search equation of observed bees in literature [11], namely:

$$v_{i,j} = \frac{x_{best,j} + x_{e,j}}{2} + \varphi_{i,j} \cdot (x_{best,j} - x_{k,j}), \tag{7}$$

Eq. (7) uses the information of the contemporary optimal solution to guide the bee's search, and it is easy to fall into the local optimal solution. However, Eq. (6) uses the information of random selection of the optimal solution of individual neighborhood to guide the search of bees, showing a certain ability of exploration.

3.4. Neighborhood Scope and Algorithm Pseudo-code

As in literature [12], we take the average Euclidean distance md as the neighborhood range of the individual. In particular, for the i-th individual, the neighborhood range is calculated as follows:

$$md_i = \frac{\sum_{j=1}^{SN} d_{i,j}}{SN - 1}, \tag{8}$$

where SN is the number of food sources; $d_{i,j}$ $(i \neq j)$ is the Euclidean distance between the food source positions x_i and x_j. If the Euclidean distance between x_j and x_i is less than the average Euclidean distance md_i, then the food source location x_j is within the neighborhood range of i -th food source.

The pseudo-code of the ERABC algorithm is provided in Algorithm 1.

Algorithm 1 ERABC Algorithm

1: Parameter initialization (population number SN, dimension D, maximum function evaluation number $\max.FES$, predetermined period $limit$)
2: According to Eq. (1), the initial population is generated and the value of its objective function is calculated.
3: **while** $FES < \max.FES$, **do**
4: Sort the individual objective function values to find the elite solution.
5: According to Eq. (8), calculate the average Euclidean distance of each individual.
6: **for** $i = 1$ to SN
7: The candidate solution v_i is generated by Eq. (5) or Eq. (2).
8: **if** $f(v_i) < f(x_i)$
9: $x_i = v_i;\ counter(i) = 0.$
10: **else**
11: $counter(i) = counter(i) + 1.$
12: **end if**
13: **end for**

ISSN: 2664-9640

14: **for** $i = 1$ to SN
15: Randomly select x_e from the elite solution.
16: The candidate solution v_e is generated by Eq. (6) or Eq. (2).
17: **if** $f(v_e) < f(x_e)$
18: $x_e = v_e$; $counter(e) = 0$.
19: **else**
20: $counter(e) = counter(e) + 1$.
21: **end if**
22: **end for**
23: $FES = FES + 2SN$.
24: **if** $counter(\max) > limit$
25: According to Eq. (1), a solution is randomly generated to replace $x_{\max}$.
26: **end if**
27: **end while**

3.5. Time Complexity Analysis

The difference between ERABC algorithm and ABC algorithm is that extra computation is introduced, which is mainly reflected in the selection of the optimal solution (population ranking) and the best neighborhood of the individual.The time complexity analysis of the algorithm in this paper is given below. In one iteration, the time complexity of selecting the elite solution (sorting the population) is $O(SN \cdot \log(SN))$. Since the complexity of ABC algorithm is $O(SN \cdot D)$, the overall complexity of sorting is $O(SN \cdot \log(SN) + SN \cdot D)$. When D is much larger than $\log(SN)$, it can be simplified to $O(SN \cdot D)$. In addition, the computational complexity of the distance between individuals and other individuals in the population is $O(SN^2 \cdot D)$. Then, each individual finds the neighbor with the smallest value of the objective function within its own neighborhood, and the complexity of this process is $O(SN^2 + SN)$. Therefore, the overall complexity of the algorithm in this paper is $O(SN^2 \cdot D)$. Moreover, the computational burden of the algorithm in this paper is mainly determined by function evaluation. Cai and Wang [13] believed that the increased computational cost of calculating the paired distance between individuals would be negligible. Therefore, when the cost function evaluation is high, the additional overhead is relatively small.

4. Simulation

Simulation experiment is carried out on 22 benchmark test functions in reference to verify the performance of the proposed algorithm [14]. Table 1 shows the search range, global optimal values, and acceptable values for each function. The comparison algorithms ABC [4], GABC [9], EABC [15] and CABC [16] are run under the environment of Windows7 64bit operating system and Matlab R2017B. All algorithms are run for 25 times independently to make a fair comparison. The relevant parameter settings of all comparison algorithms are the same as those of the original paper. The parameter settings are shown in Table 2, and $\max.FES = 5000 \cdot D$ is set as the termination condition.

ISSN: 2664-9640

Table 1. Parameter setting table of various artificial bee colony algorithms

Algorithm	parameter setting
ABC	$SN=50,\ limit=SN\cdot D$
GABC	$SN=50,\ limit=SN\cdot D$
EABC	$SN=50,\ limit=SN\cdot D,\ \mu=0.3,\ A=1,\ \delta=0.3$
CABC	$SN=50,\ limit=SN\cdot D$
ERABC	$SN=50,\ limit=SN\cdot D,\ p=0.1,\ r=1$

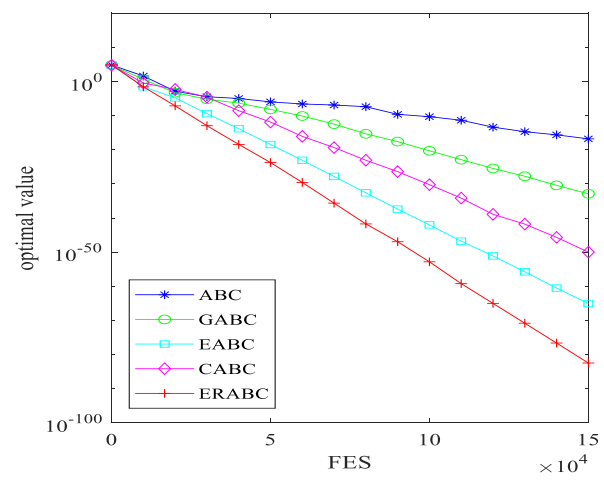
Table 2. 22 benchmark functions

Function	Name	Search space	Optimal value	Acceptable value
f_1	Sphere	$[-100,100]^D$	0	1×10^{-8}
f_2	Elliptic	$[-100,100]^D$	0	1×10^{-8}
f_3	SumSquare	$[-10,10]^D$	0	1×10^{-8}
f_4	SumPower	$[-1,1]^D$	0	1×10^{-8}
f_5	Schwefel 2.22	$[-10,10]^D$	0	1×10^{-8}
f_6	Schwefel 2.21	$[-100,100]^D$	0	1×10^{0}
f_7	Step	$[-100,100]^D$	0	1×10^{-8}
f_8	Exponential	$[-10,10]^D$	0	1×10^{-8}
f_9	Quartic	$[-1.28,1.28]^D$	0	1×10^{-1}
f_{10}	Rosenbrock	$[-5,10]^D$	0	1×10^{-1}
f_{11}	Rastrigin	$[-5.12,5.12]^D$	0	1×10^{-8}
f_{12}	NCRastrigin	$[-5.12,5.12]^D$	0	1×10^{-8}
f_{13}	Griewank	$[-600,600]^D$	0	1×10^{-8}
f_{14}	Schwefel 2.26	$[-500,500]^D$	0	1×10^{-8}
f_{15}	Ackley	$[-50,50]^D$	0	1×10^{-8}
f_{16}	Penalized 1	$[-100,100]^D$	0	1×10^{-8}
f_{17}	Penalized 2	$[-100,100]^D$	0	1×10^{-8}
f_{18}	Alpine	$[-10,10]^D$	0	1×10^{-8}
f_{19}	Levy	$[-10,10]^D$	0	1×10^{-8}
f_{20}	Weierstrass	$[-1,1]^D$	0	1×10^{-8}
f_{21}	Himmelblau	$[-5,5]^D$	-78.33236	-78
f_{22}	Michalewicz	$[0,\pi]^D$	-D	-D+1

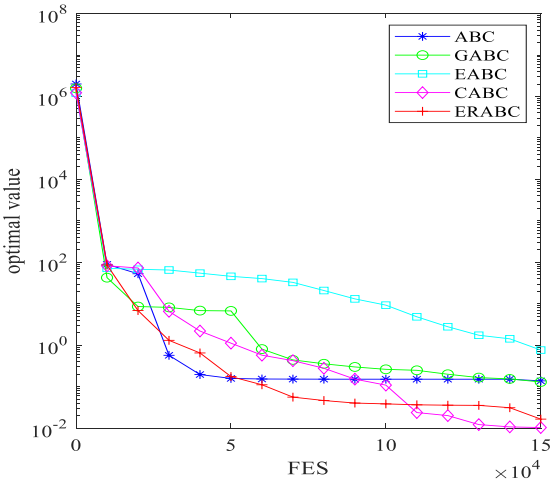
In this paper, three indexes are used to evaluate the proposed algorithm performance, that is, Mean(std) represents the average optimal objective function value of 25 independent experiments and the corresponding standard deviation, which is used to evaluate the quality of solutions obtained by different algorithms. The acceptable value is used to estimate the rate of

ISSN: 2664-9640

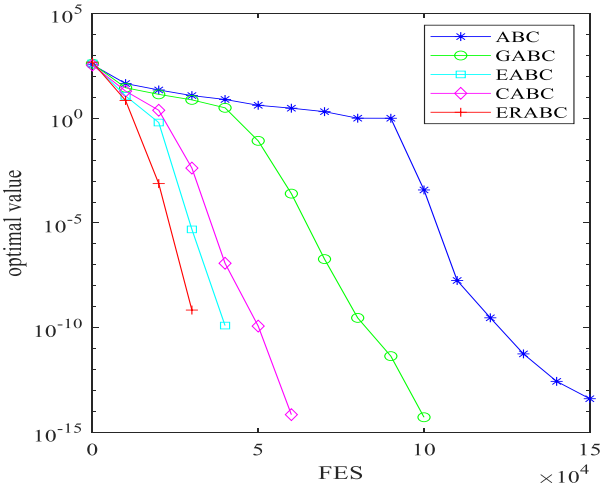
convergence, and the evaluation of the average function reaching the acceptable value is recorded with AVEN. If the algorithm cannot find an objective function solution smaller than the acceptable value, then AVEN will use NaN to represent it. The success rate (SR\%) represents the proportion of the solution whose objective function value is less than the acceptable value in 25 independent experiments, which is used to evaluate the robustness of different algorithms.



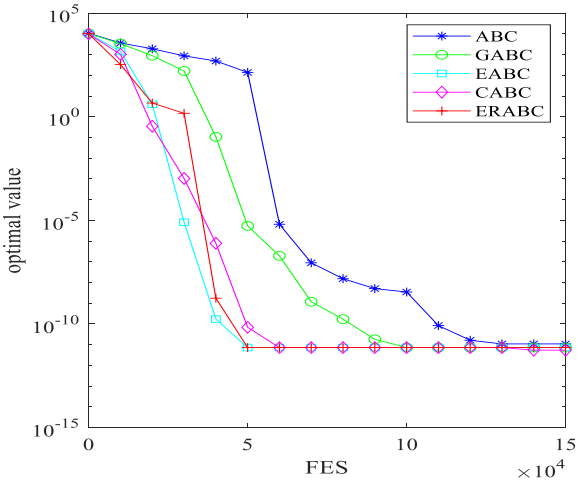
(d) Function 1



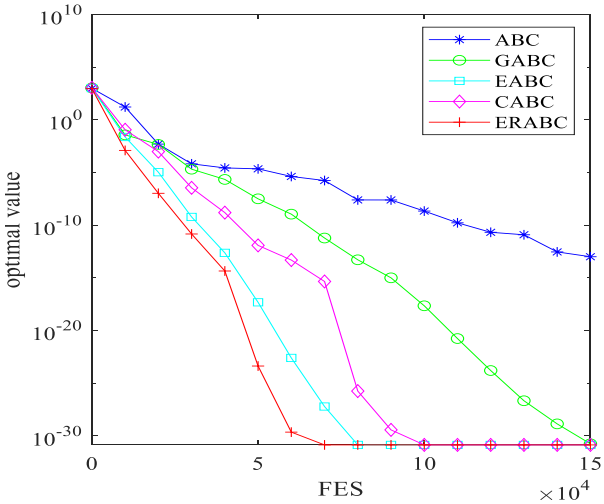
(b) Function 10



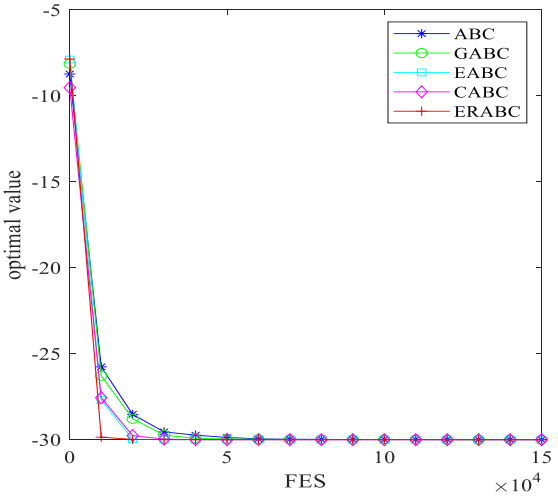
(c) Function 12



(d) Function 14



(e) Function 19



(f) Function 22

Figure 1. Comparison of convergence properties of partial functions at $D = 30$

ISSN: 2664-9640

In order to test the effectiveness of the algorithm in this paper, we give the simulation experiments on the test functions of $D=30,50,100$. Due to the limited space, we only provide the experimental results of some functions of $D=30,\ 100$. The convergence curve of the function in part $D=30$ is plotted in Figure 1 to see the change of convergence rate more clearly. Table. 3 records the experimental results when $D=100$. For the single mode functions f_1 - f_7, the solution quality and convergence speed of ERABC algorithm are superior to all comparison algorithms except f_7. Since the actual global optimal solution of f_7 is not a point, but a region, this makes it easier to solve f_7. All the algorithms have found the optimal solution, but the convergence speed in this paper is faster. Similarly, for f_8 $, other algorithms except EABC algorithm get the same result. For f_9, all the algorithms can only get an approximate global optimal solution. It is difficult to find the actual global optimal solution for f_9 because f_9 is a noisy quartic function. f_{10} is a Rosenbrock function with a variety of characteristics. Its global optimal solution is located in a narrow and long parabolic flat valley, with strong dependence among variables, and the gradient generally does not point to the optimal solution [17]. Therefore, it is easy to fall into the local optimal region by using the information of the current optimal solution or other good information. This is the reason why the algorithm in this paper does not perform well in f_{10}. For multi-modal functions f_{11} - f_{22}, from the perspective of convergence speed, EABC algorithm has a faster convergence speed than ERABC algorithm on f_{14} and f_{20}, but the success rate is not as high as ERABC algorithm. In terms of the accuracy and robustness of the solution, ERABC algorithm is superior to or at least not inferior to other algorithms in most test functions.

Table 3. The experimental results of 22 benchmark functions on 100 dimensions

Algorithm	Metric	ABC	GABC	EABC	CABC	ERABC
f_1	SR(AVEN)	**100**(293090)	**100**(178938)	**100**(103518)	**100**(142978)	**100(75730)**
f_2	SR(AVEN)	84(477721)	**100**(269998)	**100**(143454)	**100**(223362)	**100(100170)**
f_3	SR(AVEN)	**100**(279570)	**100**(175022)	**100**(100738)	**100**(223362)	**100(73782)**
f_4	SR(AVEN)	**100**(74414)	**100**(44434)	**100**(28794)	**100**(41750)	**100(21582)**
f_5	SR(AVEN)	**100**(464842)	**100**(288274)	**100**(160754)	**100**(203078)	**100(118426)**
f_6	SR(AVEN)	0(NaN)	0(NaN)	0(NaN)	0(NaN)	**100(269326)**
f_7	SR(AVEN)	**100**(52470)	**100**(45278)	**100**(30826)	**100**(34138)	**100(28358)**
f_8	SR(AVEN)	**100**(150)	**100**(150)	**100**(150)	**100**(150)	**100**(150)
f_9	SR(AVEN)	0(NaN)	0(NaN)	72(419400)	20(443790)	**100(240074)**
f_{10}	SR(AVEN)	**60**(352657)	36(388883)	28(298079)	40(343890)	48**(225692)**
f_{11}	SR(AVEN)	**100**(399922)	**100**(282398)	92(131863)	**100**(157098)	92**(120276)**
f_{12}	SR(AVEN)	80(449100)	**100**(314842)	88(144968)	96(183075)	**100(131998)**
f_{13}	SR(AVEN)	**100**(291538)	**100**(195134)	**100**(109406)	**100**(150818)	96**(91046)**
f_{14}	SR(AVEN)	**100**(307226)	**100**(266386)	76**(154418)**	**100**(167930)	88(172932)
f_{15}	SR(AVEN)	36(496128)	**100**(326006)	**100**(185154)	**100**(211510)	**100(128914)**
f_{16}	SR(AVEN)	**100**(253770)	**100**(155050)	**100**(89338)	**100**(128090)	**100(63042)**

ISSN: 2664-9640

f_{17}	SR(AVEN)	**100**(319730)	**100**(185790)	**100**(104702)	**100**(151442)	**100(73522)**
f_{18}	SR(AVEN)	0(NaN)	0(NaN)	**100(162714)**	**100**(191214)	**100**(177430)
f_{19}	SR(AVEN)	**100**(305010)	**100**(179426)	**100**(97646)	**100**(134782)	**100(78346)**
f_{20}	SR(AVEN)	0(NaN)	0(NaN)	36**(254872)**	**100**(291370)	**100**(274950)
f_{21}	SR(AVEN)	**100**(113934)	**100**(69566)	**100**(32870)	**100**(34250)	**100(27386)**
f_{22}	SR(AVEN)	**100**(156814)	**100**(143646)	**100**(77918)	**100**(86826)	**100(31934)**

As can be seen from Table 3, with the increase of dimension, various algorithms have problems to varying degrees. However, ERABC algorithm is still better than the comparison algorithm in most functions, especially for f_6 and f_9. When $D=100$, the success rate of comparison algorithm f_6 is 0%, and the success rate of ERABC algorithm is 100%. By comparison, the success rate of algorithm f_9 is low, while the success rate of ERABC algorithm reaches 100%. f_9 is a quartic function with noise, indicating that the algorithm in this paper has better robustness to f_9. However, ERABC algorithm is not suitable for f_{14}. With the increase of dimension, the success rate decreases from 100% when $D=30$ to 88% when $D=100$. For f_{10}, other algorithms and ERABC algorithm are not as robust as ABC algorithm because of the characteristics mentioned above. For some functions, ERABC algorithm and comparison algorithm get the same average value and reach a stable state because the maximum function evaluation times are relatively large, that is, the number of iterations is relatively large. However, ERABC algorithm has a faster convergence speed.

On the whole, judging from all the experimental results, each algorithm has its advantages and disadvantages. Numerical results show that the performance of the proposed algorithm is significantly superior for single mode functions, but not for some multi-mode functions. Because the search equations of GABC, EABC and ERABC algorithms all utilize the information of the current optimal solution, the results of special functions like f_{10} are not as good as those of ABC and CABC. For other individual multi-modal functions, due to the random selection of initial points in 25 independent experiments, they enter into an unpromising region after iteration, and one or two outliers may appear in the final result, thus leading to the increase of the average optimal objective function value.

5. Conclusion

In this paper, an improved artificial bee colony algorithm based on elite solution and random individual neighborhood information is proposed to improve the local search ability and convergence speed of ABC algorithm. The final simulation results show that the improved algorithm is better than the comparison algorithm. In the future, the parameter setting theory of ABC algorithm will be studied.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants no. 61871430.

ISSN: 2664-9640

References

[1] Y. Wei, C. Xu, and Q. Hu. Transformation of optimization problems in revenue management, queueing system, and supply chain management, International Journal of Production Economics, vol. 146, no. 2, pp. 588–597, 2015.

[2] G. Li and Q. Lin. A novel hybrid differential evolution algorithm with modified code and jade, Applied Soft Computing, vol. 47, pp. 577–599, 2016.

[3] M. Li, Y. Lu, and L. Jie, Hybrid multipopulation cellular genetic algorithm and its performance, Transactions of Nanjing University of Aeronautics and Astronautics, vol. 31, no. 004, pp. 405–412, 2014.

[4] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Kayseri: Engineering Faculty Computer Engineering Department, Ereiyes University, 2005.

[5] Y. Chen, L. Li, H. Peng, J. Xiao, and Y. Shi, Particle swarm optimizer with two differential mutation, Applied Soft Computing, vol. 61, pp.314–330, 2017.

[6] Y. X. A. B, Y. Z. A, and H. L. C, An elitism based multi-objective artificial bee colony algorithm, European Journal of Operational Research, vol. 245, no. 1, pp. 168–193, 2015.

[7] Ozturk, Celal, Karaboga, Dervis, Hancer, and Emrah, A novel binary artificial bee colony algorithm based on genetic operators, Information Sciences, vol. 297, pp. 154–170, 2015.

[8] X. Yan, Y. Zhu, W. Zou, and W. Liang, A new approach for data clustering using hybrid artificial bee colony algorithm, Neurocomputing, vol. 97, pp. 241–250, 2012.

[9] 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.

[10] W. Gao, F. Chan, L. Huang, and S. Liu, Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood, Information Sciences, vol. 316, pp. 180–200, 2015.

[11] L. Cui, G. Li, Q. Lin, Z. Du, W. Gao, J. Chen, and N. Lu, A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation, Information Sciences, vol. 367-368, pp.1012–1044, 2016.

[12] Q. Lin, M. Zhu, G. Li, W. Wang, L. Cui, J. Chen, and J. Lu, A novel artificial bee colony algorithm with local and global information interaction, Applied Soft Computing, vol. 62, pp. 702–735, 2017.

[13] Y. Cai and J. Wang, Differential evolution with neighborhood and direction information for numerical optimization, IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 2202–2215, 2013.

[14] W. F. Gao, L. L. Huang, S. Y. Liu, and C. Dai, Artificial bee colony algorithm based on information learning, IEEE Transactions on Cybernetics, vol. 45, no. 12, pp. 2827–2839, 2017.

[15] W. F. Gao, S. Y. Liu, and L. L. Huang, Enhancing artificial bee colony algorithm using more information-based search equations, Information Ences, vol. 270, p. 112C133, 2014.

[16] W. Gao, S. Liu, and L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Transactions on Cybernetics, vol. 43, no. 3, pp. 1011–1024, 2013.

[17] Y. W. Shang and Y. H. Qiu, A note on the extended rosenbrock function, Evolutionary Computation, vol. 14, no. 1, pp. 119–126, 2006.