Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2024, 11(4):164-169

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

An Improved Artificial Bee Colony Algorithm for Numerical Function Optimization

Zhigang Wang

School of Mathematics, Nanjing Normal University, Taizhou College, Jiangsu

Abstract Artificial bee colony (ABC) algorithm is a recently introduced swarm based meta-heuristic algorithm. ABC algorithm simulates the intelligent foraging behavior of honey bee swarm. However, there is still an insufficiency in ABC algorithm regarding its solution search equation, which is good at exploration but poor at exploitation. To remedy this problem, an improved artificial bee colony algorithm is proposed. In our method, the employed bees and the onlooker bees use different solution search equation to maintain the balance between exploitation and exploration. The comparison results on a set of 10 benchmark functions demonstrate the proposed method has fast convergence and high accuracy than other ABC-based algorithms.

Keywords Swarm intelligence; Artificial bee colony algorithm; Search equation

1. Introduction

In recent decades, many swarm intelligence-based heuristic optimization techniques such as ant colony optimization (ACO) [1], particle swarm optimization (PSO) [2], and artificial bee colony (ABC) algorithm [3] have been proposed in the literature. These algorithms simulate the collective behavior of groups of simple agents like colonies of ants, flocks of birds, swarm of bees, and so on. ABC algorithm is a biologically inspired population-based meta-heuristic algorithm that mimics the foraging behavior of honey bee swarms. Due to its simplicity and ease of implementation, ABC algorithm has captured much attention and has been widely used to solve many practical optimization problems [4-9]. Karaboga et al. [10, 11] analyzed the performance of ABC algorithm by comparing it to other novel evolutionary algorithms on well-known benchmark problems. Results show that ABC algorithm is superior to other meta-heuristic algorithms such as genetic algorithm (GA), differential evolution (DE), and PSO, on most of the instances.

However, the solution search equation of ABC algorithm is good at exploration but poor at exploitation [12], which results in slow convergence. In order to improve the performance of the ABC algorithm, a number of variant ABC algorithms have been developed. For example, Gao et al. [13] proposed two ABC-based algorithms that use two update rules of differential evolution called ABC/Best/1 and ABC/Best/2. Gao and Liu [14] used the update rule of the ABC/Best/1 algorithm for employed bees and the update rule of basic ABC algorithm for onlooker bees to reinforce the exploration ability of the method. Akay and Karaboga [15] proposed a modified ABC algorithm which employs the frequency of perturbation and the ratio of the variance operation. Banharnsakun et al. [16] described best-so-far selection in ABC algorithm where the best feasible solutions found so far are used globally among the entire population. Liu et al. [17] presented an improved version of ABC algorithm in which the best-so-far information, inertial weight and acceleration coefficient are used when employing local search.

In this paper, an improved artificial bee colony algorithm is proposed. In our method, the employed bees and the onlooker bees use different solution search equation to maintain the balance between exploitation and exploration. The new artificial bee colony algorithm is called IABC algorithm for short. Our experiment results tested on a set of 10 benchmark functions show that IABC algorithm has fast convergence and high accuracy than other ABC-based algorithms.

The remainder of this paper is organized as follows. Section 2 describes the original ABC algorithm. Section 3 introduces the ABC algorithm with modified search equation. Section 4 presents and discusses the experimental results; Finally, Section 5 offers our conclusions.

2 Artificial Bee Colony Algorithm

ABC algorithm is a relatively new swarm intelligence algorithm based on the foraging behavior of honey bee swarms. The colony of artificial bees in the ABC algorithm consists of three groups of bees: employed bees, onlooker bees and scout bees. Employed bees are responsible for search of a food source and for sharing this information to recruit onlooker bees. Onlooker bees tend to select better food sources from those employed bees, and further search the food around the selected food source. If a food source is not improved by a predetermined number of trials (denoted by *limit*), this employed bee will become a scout bee to search randomly for new food source.

In ABC algorithm, a food source position represents a potential solution to the problem to be optimized, and the nectar amount of each food source is the fitness of the corresponding solution. The number of employed bees or onlooker bees is equal to the number of solutions in the population. First, let us suppose the solution space of the problem is D-dimensional. The ABC algorithm starts with randomly producing food source, and each solution is represented as $x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$, $i \in \{1, 2, \cdots, SN\}$, SN is equals to the number of food sources and half the population size. Each food source is generated as follows:

$$x_i^j = x_{\min}^j + rand\,(x_{\max}^j - x_{\min}^j) \text{`}\backslash \qquad (1)$$

where $i = 1, 2, \cdots, SN$, $j = 1, 2, \cdots, D$. $x_{\min}^j$ and $x_{\max}^j$ are the lower and upper bounds for the dimension j, respectively. $rand$ is a random number in the range $[0,1]$

After initialization, each employed bee performs a modification on the position of the food source by randomly selecting a neighboring food source. A new food source v_i can be generated from the old food source as follows:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \qquad (2)$$

where $j \in \{1, 2, \cdots, D\}$ and $k \in \{1, 2, \cdots, SN\}$ are randomly chosen indexes; k is different from i; φ_{ij} is a random real number within the range $[-1,1]$.

Once v_i is obtained, it will be evaluated and compared to x_i. If the fitness of v_i is equal to or better than that of x_i, v_i will replace x_i and become a new member of the population. Otherwise, x_i is retained.

After all employed bees complete the search process, they share their information related to the nectar amounts and the positions of their sources with onlooker bees. Each onlooker bee chooses a better performed food source solution from all the food source solutions of the employed bees. The onlooker bee selects a food source solution depend on the roulette wheel selection mechanism, which is given by

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad (3)$$

$$fit_i = \begin{cases} \dfrac{1}{1+f_i}, & f_i \geq 0 \\ 1+|f_i|, & f_i < 0 \end{cases} \qquad (4)$$

where fit_i is the fitness value of the solution x_i. After the selection, the onlooker bee selects her food source x_i, she produces a modification on x_i by Eq. (2). If the modified food source has a better or equal nectar amount than x_i, the modified food source will replace x_i and become a new solution in the population.

If the food source x_i cannot be further improved through a given number of steps (*limit*) in the ABC algorithm, the food source will be abandoned, and the corresponding employed bee becomes a scout. The scout bee generates a new food source by Eq. (1).

3 Improved Artificial Bee Colony Algorithm

It is well known that the compromise between exploration and exploitation throughout the course of a run is critical to the success of an evolutionary algorithm. In the optimization algorithms, the exploration refers to the ability to investigate the various unknown regions in the solution space to discover the global optimum. The exploitation refers to the ability to apply the knowledge of the previous good solutions to find better solutions. In practice, the exploration and the exploitation contradict with each other, and in order to achieve good optimization performance, the two abilities should be well balanced. However, according to the solution search equation of ABC algorithm described by Eq. (2), the coefficient φ_{ij} is a random number in the range $[-1,1]$ and x_{kj} is a random individual in the population, therefore, the solution search dominated by Eq. (2) is random enough for exploration. In other words, the solution search equation described by Eq. (2) is good at exploration but poor at exploitation. In addition, the employed bees and the onlooker bees use the same solution search equation, this way contradict with the foraging behavior of honey bee swarm. To remedy these deficiencies, many researchers have proposed some modified solution search equations as follows:

$$v_{ij} = x_{rj} + \varphi_{ij}(x_{rj} - x_{kj}) \tag{5}$$

$$v_{ij} = x_{best,j} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{6}$$

$$v_{ij} = x_{best,j} + \varphi_{ij}(x_{rj} - x_{kj}) \tag{7}$$

where x_r is the solution which is randomly selected from the population, r and k are mutually exclusive integers randomly chosen from $\{1,2,\cdots,SN\}$, and both of them are different from the base index i, x_{best} is the global best solution of the population. To increase diversity in the population, the random neighbor is selected in Eq. (5) [18]. To support the local search around the global best solution of the population and to provide fast convergence to optimum or near optimum solutions, Eq. (6) [19] and Eq. (7) [14] are used.

Based on the above improved search equations, we propose a novel solution search equations for employed bees as follows:

$$v_{ij} = s_1 x_{ij} + (1-s_1)x_{rj} + \varphi_{ij}\left[s_1(x_{ij} - x_{kj}) + (1-s_1)(x_{rj} - x_{kj})\right] \tag{8}$$

$$s_1 = \begin{cases} 0, & rand1 < 0.5 \\ 1, & otherwise \end{cases}$$

where $rand1$ is a random number in the range $[0,1]$.

The solution search equation for onlooker bees as follows:

$$v_{ij} = x_{gbest,j} + \varphi_{ij}\left[s_2(x_{ij} - x_{kj}) + (1-s_2)(x_{rj} - x_{kj})\right] \tag{9}$$

$$s_2 = \begin{cases} 0, & rand2 < 0.5 \\ 1, & otherwise \end{cases}$$

where $rand2$ is a random number in the range $[0,1]$

4 Experiments and Comparisons

4.1 Benchmark functions and parameter settings

Table 1: Benchmark functions used in experiments

Function	Search range	Min
$f_1(x)=\sum_{i=1}^{D} x_i^2$	[-100,100]	0
$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[-10,10]	0
$f_3(x)=\max_i\{\lvert x_i\rvert, 1\le i\le D\}$	[-100,100]	0
$f_4(x)=\sum_{i=1}^{D}\left(\lfloor x_i+0.5\rfloor\right)^2$	[-100,100]	0
$f_5(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2]$	[-10,10]	0
$f_6(x)=\sum_{i=1}^{D} i x_i^4+\text{random}[0,1)$	[-1.28,1.28]	0
$f_7(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$	[-5.12,5.12]	0
$f_8(x)=\frac{1}{4000}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	[-600,600]	0
$f_9(x)=20+e-20\exp\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\}-\exp\{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\}$	[-32,32]	0
$f_{10}(x)=0.5+\frac{\sin^2\left(\sqrt{\sum_{i=1}^{D} x_i^2}\right)-0.5}{\left(1+0.001\sum_{i=1}^{D} x_i^2\right)^2}$	[-100,100]	0

In this section, IABC algorithm is applied to optimize 10 scalable benchmark functions with $D=30$, as shown in Table 1. The set of experiments tested on 10 numerical benchmark functions are performed to compare the performance of IABC algorithm with that of ABC algorithm. In all simulations, the population size of ABC algorithm and IABC algorithm is 40 (i.e., $SN=20$), $\lim it$ is $SN*D$, the number of maximum function evaluations is set to 150000.

4.2 Experimental results

Table 2: Performance comparisons of ABC and IABC

Fun	Algorithm	Best	Worst	Mean	Std
f_1	ABC	2.75E-16	6.95E-16	4.93E-16	7.98E-17
	IABC	1.45E-106	3.29E-103	1.86E-104	5.92E-104
f_2	ABC	9.86E-16	1.63E-15	1.31E-15	1.54E-16
	IABC	5.18E-55	1.63E-53	4.10E-54	3.86E-54
f_3	ABC	2.80E-01	2.39E+00	8.37E-01	4.72E-01
	IABC	4.40E-02	1.40E-01	7.58E-02	2.47E-02
f_4	ABC	0	0	0	0
	IABC	0	0	0	0
f_5	ABC	4.23E-04	1.74E-01	4.32E-02	4.51E-02
	IABC	8.50E-04	7.86E-01	1.14E-01	1.86E-01
f_6	ABC	1.96E-02	7.11E-02	4.85E-02	1.29E-02
	IABC	1.11E-02	2.56E-02	1.82E-02	3.61E-03
f_7	ABC	0	0	0	0
	IABC	0	0	0	0
f_8	ABC	0	4.88E-06	1.78E-07	8.89E-07
	IABC	0	0	0	0
f_9	ABC	2.55E-14	4.32E-14	3.55E-14	3.62E-15
	IABC	1.48E-14	2.90E-14	2.29E-14	3.77E-15
f_{10}	ABC	2.28E-01	3.96E-01	3.18E-01	5.19E-02
	IABC	1.27E-01	3.12E-01	2.18E-01	5.64E-02

Table 2 show the comparison results in terms of the best, worst, mean and standard deviation (std) of the solutions obtained in the 30 independent runs by each algorithm. As is shown in Table 2, when solving the unimodal functions, an interesting result we notice is that two algorithms have most reliably found the minimum of f_4. It is a region rather than a point in f_4 that is the optimum. Hence, this problem may relatively be easy to solve with a 100% success rate. For other unimodal functions, these results indicate that IABC algorithm offers the higher accuracy on almost all the functions except functions f_5. This is because f_5 is a relatively hard test problem, it has a very narrow valley from local optimum to global optimum. When solving the multimodal functions, IABC algorithm can find the optimal or closer-to-optimal solutions on the complex multimodal functions f_7, f_8 and f_9.

The further experiment results are listed in Table 3, which show the performance comparisons among GABC algorithm [12], ABCbest1 algorithm [13], ABCbest2 algorithm [13], MABC algorithm [14] and IABC algorithm on 10 benchmark functions. The number of maximum function evaluations is set to 150000, and the best results are marked in bold. The results, which have been summarized in Table 3, show that IABC algorithm performs much better in most cases than these ABC variants.

Table 3: Performance comparisons of GABC, ABCBest1, ABCBest2, MABC and IABC

Fun	Metric	GABC	ABCBest1	ABCBest2	MABC	IABC
f_1	Mean	4.62E-16	3.11E-47	5.96E-35	9.43E-32	**1.86E-104**
	Std	7.12E-17	3.44E-47	3.61E-35	6.67E-32	**5.92E-104**
f_2	Mean	1.35E-15	2.10E-25	1.36E-18	2.40E-17	**4.10E-54**
	Std	1.36E-16	9.08E-26	4.27E-19	9.02E-18	**3.86E-54**
f_3	Mean	2.18E-01	2.18E+00	3.55E+00	1.02E+01	**7.58E-02**
	Std	4.01E-02	3.27E-01	4.79E-01	1.49E+00	**2.47E-02**
f_4	Mean	**0**	**0**	**0**	**0**	**0**
	Std	**0**	**0**	**0**	**0**	**0**
f_5	Mean	3.21E-01	1.49E+01	5.45E+00	6.11E-01	**1.14E-01**
	Std	8.21E-01	2.87E+01	8.40E+00	4.55E-01	**1.86E-01**
f_6	Mean	2.03E-02	2.06E-02	2.53E-02	3.71E-02	**1.82E-02**
	Std	5.74E-03	4.75E-03	4.67E-03	8.53E-03	**3.61E-03**
f_7	Mean	**0**	**0**	**0**	**0**	**0**
	Std	**0**	**0**	**0**	**0**	**0**
f_8	Mean	3.70E-17	**0**	1.81E-08	**0**	**0**
	Std	5.32E-17	**0**	6.29E-08	**0**	**0**
f_9	Mean	3.20E-14	3.01E-14	3.07E-14	4.13E-14	**2.29E-14**
	Std	3.36E-15	2.91E-15	3.43E-15	2.17E-15	**3.77E-15**
f_{10}	Mean	2.66E-01	2.39E-01	2.81E-01	2.95E-01	**2.18E-01**
	Std	4.39E-02	6.13E-02	3.92E-02	3.17E-02	**5.64E-02**

Summarizing the earlier statements, it can be concluded IABC algorithm can improve bees' searching abilities, not only avoid falling into local optimum when solving the multimodal functions, but also increase the convergence speed and compute more efficiently when dealing with the complex unimodal functions.

5 Conclusion

In this paper, an improved artificial bee colony algorithm is presented, called IABC algorithm. In IABC algorithm, the employed bees and the onlooker bees use different solution search equation to maintain the balance between exploitation and exploration. We testify the performance of the IABC algorithm on a suite of unimodal/multimodal benchmark functions and provide comparisons with some other algorithms. The results show that the IABC algorithm possesses superior performance in accuracy, convergence speed, stability and robustness, as compared to the other algorithms. Hence, the IABC algorithm may be a promising and viable tool to deal with complex numerical optimization problems.

References

[1]. M. Dorigo, V. Maniezzo, A. Colorni. The ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics Part B-Cybernetics, 26(1), (1996), 1-13.

[2]. J. Kennedy, R.C. Eberhart. Particle swarm optimization, in: Proc. of IEEE International Conference on Neural Networks, Piscataway, USA, 1995, 1942-1948.
[3]. D. Karaboga, B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization, 39(3), 2007, 459-471.
[4]. A. Singh. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, Applied Soft Computing, 9(2), 2009, 625-631.
[5]. D. Karaboga, C. Ozturk. A novel clustering approach: artificial bee colony (ABC) algorithm, Applied Soft Computing, 11(1), 2011, 652-657.
[6]. D. Karaboga, S. Okdem, C. Ozturk. Cluster based wireless sensor network routing using artificial bee colony algorithm, Wireless Networks, 18(7), 2012, 847-860.
[7]. V.J. Manoj, E. Elias. Artificial bee colony algorithm for the design of multiplier-less nonuniform filter bank transmultiplexer, Information Sciences, 192(6), 2012, 193-203.
[8]. W. C. Yeh, T. J. Hsieh. Solving reliability redundancy allocation problems using an artificial bee colony algorithm, Computers & Operations Research, 38(11), 2011, 1465-1473.
[9]. W. Y. Szeto, Y. Wu, S. C. Ho. An artificial bee colony algorithm for the capacitated vehicle routing problem, European Journal of Operational Research, 215(1), 2011, 126-135.
[10]. D. Karaboga, B. Basturk. On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing, 8(1), 2008, 687-697.
[11]. D. Karaboga, B. Akay. A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation, 214(1), 2009, 108-132.
[12]. G. Zhu, S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization, Applied Mathematics and Computation, 217(7), 2010, 3166-3173.
[13]. W. Gao, S. Liu, L. Huang. A global best artificial bee colony algorithm for global optimization, Journal of Computational and Applied Mathematics, 236(11), 2012, 2741-2753.
[14]. W. Gao, S. Liu. A modified artificial bee colony algorithm, Computer & Operations Research, 39(3), 2012, 687-697.
[15]. B. Akay, D. Karaboga. A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences, 192(6), 2012, 120-142.
[16]. A. Banharnsakun, T. Achalakul, B. Sirinaovakul. The best-so-far selection in artificial bee colony algorithm, Applied soft Computing, 11(2), 2011, 2888–2901.
[17]. G. Liu, P. Niu, X. Xiao. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization, Applied soft Computing, 12(1), 2012, 320-332.
[18]. W. Gao, S. Liu, L. Huang. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Transactions on Cybernetics, 43(3), 2013, 1011-1024.
[19]. W. Gao, S. Liu. Improved artificial bee colony algorithm for global optimization, Information Processing Letters, 111(17), 2011, 687-697.