Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2017, 4(3):37-43**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Modified Artificial Bee Colony Algorithm for Numerical Function Optimization

Wang Zhi-gang

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

Abstract Artificial bee colony (ABC) algorithm is a relatively new optimization algorithm inspired by the foraging behavior of honey bee swarm, which has been used to solve many real-world optimization problems, and it performs well in most cases. However, ABC algorithm does well in exploration but badly in exploitation. In order to balance the exploration and the exploitation, a modified artificial bee colony algorithm is proposed. In this method, the employed bees search only around the random solution which is selected from the population to improve exploration, and the onlooker bees search only around the solution which is selected from the population by using the roulette wheel to improve exploitation. In addition, we use a more robust calculation to determine and compare the quality of alternative solutions. 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 Artificial bee colony algorithm; Search equation; Optimization

Introduction

People have developed several kinds of biological-inspired optimization algorithms to solve complicated problems in recent decades, such as ant colony optimization (ACO) [1] inspired by the social behavior of ants in an ant colony, particle swarm optimization (PSO) [2] inspired by the behaviors of bird flocks and fish schools, and artificial fish swarm algorithm (AFSA) [3] inspired by the collective movement of the fish and their various social behaviors. By simulating the foraging behavior of honey bee swarm, Karaboga [4] invented a new kind of optimization algorithm called artificial bee colony (ABC) algorithm. A set of experimental results on function optimization [5-8] show that ABC algorithm is superior to other optimization algorithms such as genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO), on most of the instances. Due to its simplicity and ease of implementation, researchers have successfully applied ABC algorithm to solve both continuous and discrete optimization problems since its invention [9-15].

However, like other evolutionary algorithms, ABC algorithm also faces slow convergence when handing the unimodal problems and easily get trapped in the local optima when solving complex multimodal problems. To solve these issues, a number of variant ABC algorithms have been proposed. For example, Zhu and Kwong [16] proposed a global-best guided ABC algorithm, which utilizes the global best individual's information within the search equation to improve the exploitation. Gao *et al*. [17] proposed two ABC-based algorithms that use two update rules of differential evolution called ABC/Best/1 and ABC/Best/2. Gao and Liu [18] 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 [19] introduced a modified version of the ABC in which frequency of perturbation is controlled adaptively and the ratio of variance operator was introduced. Banharnsakun et al. [20] described best-so-far selection in ABC algorithm where the best feasible solutions found so far are used globally among the entire population.

In this paper, we propose a modified artificial bee colony algorithm. In this method, the employed bees search only around the random solution which is selected from the population to improve exploration, and the onlooker bees search only around the solution which is selected from the population by using the roulette wheel to improve exploitation. We name the modified artificial bee colony algorithm as MABC algorithm for short. In addition, the comparison and the selection of the new solution are changed from a fitness-based comparison to an objective-value based.

The rest of this paper is organized as follows: Section 2 describes the original ABC algorithm; Section 3 presents the modified ABC algorithm; the computational experiments and results are presented in Section 4 and finally Section 5 concludes the paper.

Artificial Bee Colony Algorithm

ABC algorithm is a relatively new optimization algorithm that mimics the foraging behavior of honey bee swarms. In ABC algorithm, the colony of artificial bees consists of three groups, namely, employed bees, onlooker bees and scout bees. Employed bees search the food around the food source and come back to hive and exchange the information with onlooker bees by dancing on the dance area. Onlooker bees tend to select good food sources from those founded by the employed bees, then further search the foods around the selected food source. The employed bee which food sources have been abandoned becomes a scout bee and starts searching for a new food source.

For the purpose of optimization, the position of the food source represents a possible solution to the optimization problem, 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. For the first step, the ABC algorithm generates a randomly distributed population of SN solutions (food source), where SN also represents the number of employed bees or onlooker bees. Each solution $x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$ represents the i th food source in the population, where D is the number of parameters to be optimized. Each food source is generated as follows:

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

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. $rand(0,1)$ is a random real number within the range $[0,1]$.

After initialization, each employed bee x_i generates a new food source v_i in the neighborhood of its present position by using solution search equation as follows:

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

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. An onlooker bee evaluates the nectar information and chooses a food source with a probability P_i related to its nectar amount. The value of P_i is calculated for the i th food source as follows:

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

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

where $i = 1, 2, \cdots, SN$ and fit_i is the fitness value of the solution x_i. By using this mechanism, these food sources that have better fitness values will be more likely to be selected for update. After 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 a food source x_i cannot be improved for a predetermined number of $\lim it$, the food source will be abandoned, and the employed bee associated with that food source becomes a scout bee. Then, the scout bee finds a new food source using Eq. (1).

3. Modified Artificial Bee Colony Algorithm

As can be seen from Eq. (2), the coefficient φ_{ij} is a random real number within 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. In order to solve these issues, we propose a novel solution search equations for onlooker bees as follows:

$$v_{ij} = x_{mj} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (5)$$

where x_m is the solution which is selected from the population by using the roulette wheel, k is a random integer selected from $\{1, 2, \cdots, SN\}$, and different from the base index i, φ_{ij} is a random real number within the range $[-1,1]$.

To increase diversity in the population and increase the exploration, the solution search equation for employed bees as follows:

$$v_{ij} = x_{rj} + \varphi_{ij}(x_{rj} - x_{kj}) \quad (6)$$

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.

In ABC algorithm, we also focus the method that is used to compare and to select between the old solution and the new solution. Basically, the comparison of the new solution and the old solution is done by the fitness value. If the fitness of the new solution is better than the fitness of the old solution, we select the new solution and ignore the old solution. However, this method has shortcoming [20]. In this paper, we use the objective function value for comparison and selection of the better solution.

Numerical Experiments and Results

In this section, MABC algorithm is applied to optimize 10 scalable benchmark functions with $D = 30$, as shown in Table 1. These functions have different properties such as unimodality, multimodality, separable and non-separable. These properties of the functions are given in column C of Table 1. In column C of Table 1, U shows that the function is unimodal, M shows that the function is multimodal, S shows that the function is separable and N shows that the function is non-separable. These unimodal functions are used to test the exploitation ability of the algorithm, and multimodal functions are used to test the exploration ability of the algorithm.

Table 1: Benchmark functions used in experiments

Function name	Function	C	Search range	Min
Sphere	$f_1(x)=\sum_{i=1}^{D} x_i^2$	US	[-100,100]	0
Schwefel 2.22	$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	UN	[-10,10]	0
Schwefel 2.21	$f_3(x)=\max_i\{\lvert x_i\rvert, 1\le i\le D\}$	UN	[-100,100]	0
Step	$f_4(x)=\sum_{i=1}^{D}\left(\lfloor x_i+0.5\rfloor\right)^2$	US	[-100,100]	0
Rosenbrock	$f_5(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2]$	UN	[-10,10]	0
Quartic	$f_6(x)=\sum_{i=1}^{D} i x_i^4+\text{random}[0,1)$	US	[-1.28,1.28]	0
Rastrigin	$f_7(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$	MS	[-5.12,5.12]	0
Griewank	$f_8(x)=\frac{1}{4000}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	MN	[-600,600]	0
Ackley	$f_9(x)=20+e-20\exp\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\}-\exp\{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\}$	MN	[-32,32]	0
Schaffer	$f_{10}(x)=0.5+\frac{\sin^2\left(\sqrt{\sum_{i=1}^{D}x_i^2}\right)-0.5}{\left(1+0.001\sum_{i=1}^{D}x_i^2\right)^2}$	MN	[-100,100]	0

In order to testify the efficiency of MABC algorithm, the experiment results are compared with the original ABC algorithm. We set the maximum number of function evaluations to be 150,000 for each function. The population size of ABC algorithm and MABC algorithm is 40 (i.e., $SN=20$), $\lim it$ is $SN*D$.

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. To vividly describe the advantage of the proposed method, the convergence curves of some test functions are plotted in Fig.1.

Table 2: Performance comparisons of ABC and MABC

Function	Algorithm	Best	Worst	Mean	Std
f_1	ABC	2.74E-16	7.44E-16	5.31E-16	9.48E-17
	MABC	5.45E-114	1.85E-109	2.40E-110	4.60E-110
f_2	ABC	1.17E-15	1.44E-15	1.32E-15	1.09E-16
	MABC	9.05E-58	6.69E-55	3.05E-56	1.21E-55
f_3	ABC	2.55E-01	1.65E+00	7.89E-01	3.67E-01
	MABC	3.87E-02	8.08E-02	5.53E-02	1.03E-02
f_4	ABC	0	0	0	0
	MABC	0	0	0	0
f_5	ABC	4.23E-04	1.74E-01	4.32E-02	4.51E-02
	MABC	9.93E-05	4.69E+00	3.06E-01	8.68E-01
f_6	ABC	2.32E-02	7.51E-02	4.83E-02	1.15E-02
	MABC	8.15E-03	2.26E-02	1.60E-02	4.03E-03
f_7	ABC	0	0	0	0
	MABC	0	0	0	0
f_8	ABC	0	6.64E-11	2.30E-12	1.19E-11
	MABC	0	2.41E-11	8.04E-13	4.33E-12
f_9	ABC	2.55E-14	4.32E-14	3.46E-14	4.37E-15
	MABC	1.84E-14	2.90E-14	2.34E-14	3.03E-15
f_{10}	ABC	2.28E-01	3.96E-01	3.20E-01	4.24E-02
	MABC	1.27E-01	3.46E-01	2.29E-01	4.94E-02

As is shown in Table 2, when solving the unimodal functions, an interesting result 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, MABC algorithm offers the higher accuracy on almost all the functions except functions f_5. When solving the multimodal functions, MABC algorithm can find the optimal or closer-to-optimal solutions on the complex functions f_7, f_8 and f_9. As can be seen from Fig.1, Compare with ABC algorithm, MABC algorithm has faster convergence speed and higher calculate accuracy.

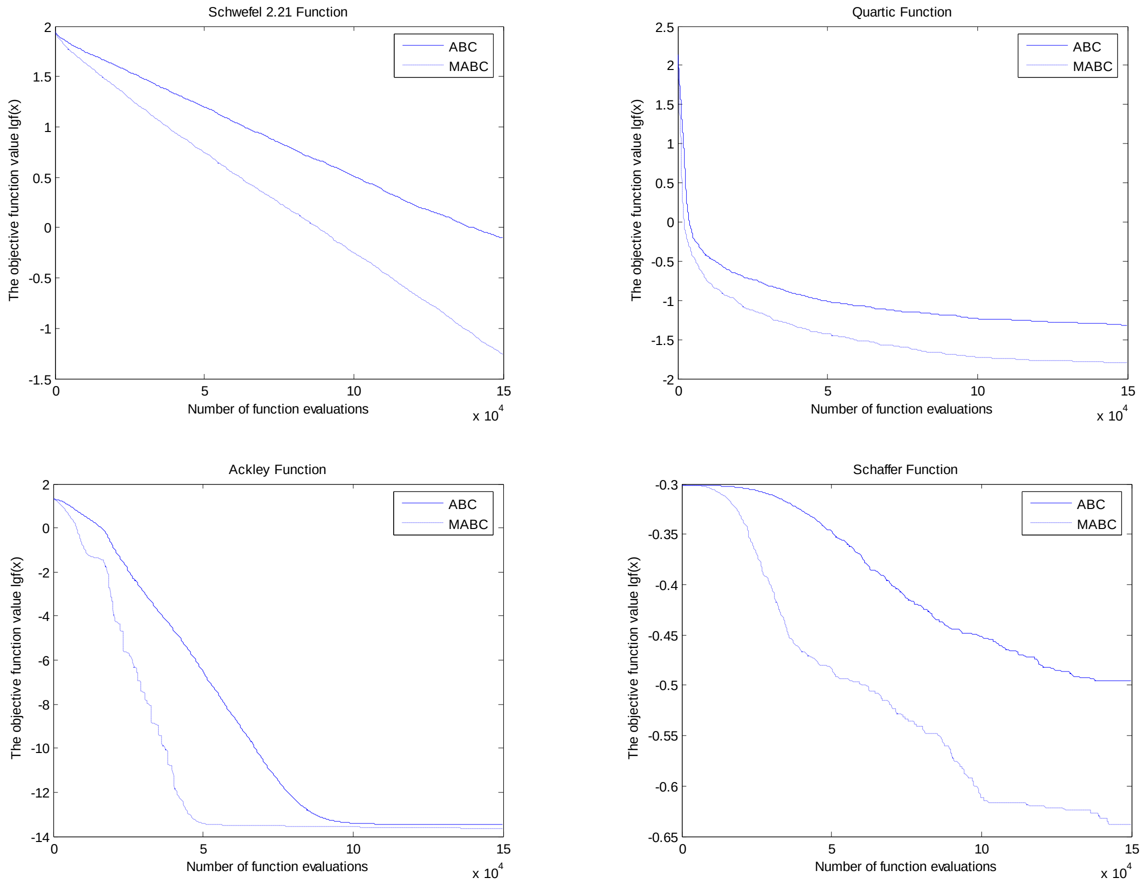

Figure 1: *Convergence curves of ABC and MABC on the 4 test functions*

In Table 3, MABC algorithm is further compared with ABCBest1 [17] and ABCBest2 [17], the number of maximum function evaluations is set to 150000. The results show that MABC algorithm performs much better in most cases than these ABC variants.

Table 3: Performance comparisons of ABCBest1, ABCBest2 and MABC

Function	ABCBest1		ABCBest2		MABC	
	Mean	Std	Mean	Std	Mean	Std
f_1	3.11E-47	3.44E-47	5.96E-35	3.61E-35	2.40E-110	4.60E-110
f_2	2.10E-25	9.08E-26	1.36E-18	4.27E-19	3.05E-56	1.21E-55
f_3	2.18E+00	3.27E-01	3.55E+00	4.79E-01	5.53E-02	1.03E-02
f_4	0	0	0	0	0	0
f_5	1.49E+01	2.87E+01	5.45E+00	8.40E+00	3.06E-01	8.68E-01

f_6	2.06E-02	4.75E-03	2.53E-02	4.67E-03	1.60E-02	4.03E-03
f_7	0	0	0	0	0	0
f_8	0	0	1.81E-08	6.29E-08	8.04E-13	4.33E-12
f_9	3.01E-14	2.91E-15	3.07E-14	3.43E-15	2.34E-14	3.03E-15
f_{10}	2.39E-01	6.13E-02	2.81E-01	3.92E-02	2.29E-01	4.94E-02

From the above statements, it can be concluded MABC algorithm can not only avoid falling into local optimum when solving the multimodal functions, but also improve the convergence speed when dealing with the complex unimodal functions.

Conclusion

This study presents a modified artificial bee colony algorithm, called MABC algorithm. In MABC algorithm, the employed bees search only around the random solution which is selected from the population, and the onlooker bees search only around the solution which is selected from the population by using the roulette wheel. At the same time, the comparison and the selection of the new solution are changed from a fitness-based comparison to an objective-value based. The proposed MABC algorithm is tested on ten different benchmark problems and compared to other well-known variants of the ABC algorithm. The comparison results demonstrate that the MABC algorithm outperforms the original ABC algorithm and other variants of the ABC algorithm in terms of solution quality and robustness for most of the experiments.

References

[1]. M. Dorigo, V. Maniezzo, A. Colorni. (1996). The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B-Cybernetics*, 26(1): 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]. X. L. Li, Z. J. Shao, J. X. Qian. (2002). An optimizing method based on autonomous animates: fish-swarm algorithm. *System Engineering-Theory & Practice*, 22(11): 32-38.

[4]. D. Karaboga, B. Basturk. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3): 459-471.

[5]. B. Basturk, D. Karaboga. (2006). An artificial bee colony (ABC) algorithm for numeric function optimization. In: IEEE Swarm Intelligence Symposium.

[6]. D. Karaboga, B. Basturk. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3): 459-471.

[7]. D. Karaboga, B. Basturk. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1): 687-697.

[8]. D. Karaboga, B. Akay. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1): 108-132.

[9]. A. Singh. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, 9(2): 625-631.

[10]. D. Karaboga, C. Ozturk. (2011). A novel clustering approach: artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 11(1): 652-657.

[11]. H. C. Tsai. (2014). Integrating the artificial bee colony and bees algorithm to face constrained optimization problems. *Information Sciences*, 258(3): 80-93.

[12]. D. Karaboga, S. Okdem, C. Ozturk. (2012). Cluster based wireless sensor network routing using artificial bee colony algorithm. *Wireless Networks*, 18(7): 847-860.

[13]. V.J. Manoj, E. Elias. (2012). Artificial bee colony algorithm for the design of multiplier-less nonuniform filter bank transmultiplexer. *Information Sciences*, 192(6): 193-203.
[14]. W. C. Yeh, T. J. Hsieh. (2011). Solving reliability redundancy allocation problems using an artificial bee colony algorithm. *Computers & Operations Research*, 38(11): 1465-1473.
[15]. W. Y. Szeto, Y. Wu, S. C. Ho. (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research*, 215(1): 126-135.
[16]. G. Zhu, S. Kwong. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7): 3166-3173.
[17]. W. Gao, S. Liu, L. Huang. (2012). A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics*, 236(11): 2741-2753.
[18]. W. Gao, S. Liu. (2012). A modified artificial bee colony algorithm. *Computer & Operations Research*, 39(3): 687-697.
[19]. B. Akay, D. Karaboga. (2012). A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 192(6): 120-142.
[20]. A. Banharnsakun, T. Achalakul, B. Sirinaovakul. (2011). The best-so-far selection in artificial bee colony algorithm. *Applied soft Computing*, 11(2): 2888–2901.