Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2020, 7(10):56-60

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

An Improved Artificial Bee Colony Algorithm for Global Optimization

Wang Zhi-gang*

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

Abstract Artificial bee colony (ABC) algorithm is a stochastic optimization algorithm based on swarm intelligence, which is inspired by the intelligent foraging behavior of bees. It has been proved to be a very effective algorithm for solving complex optimization problems. However, Basic ABC algorithm has poor exploitation ability and slow convergence speed. To solve these issues, an improved artificial bee colony algorithm is proposed in this paper. The employed bees and the onlooker bees utilize the best solution, the second and third best solutions of the whole population to generate candidate food source. This method can enhance the exploitation ability and accelerate the convergence speed. The simulation results of 10 benchmark functions demonstrate that the proposed algorithm is superior to the comparison algorithm on most test functions.

Keywords Artificial bee colony algorithm; Optimization; Benchmark function

1. Introduction

With the development of science and technology, the optimization problem has become more and more complicated. To solve these complex problems, many researchers have proposed lots of intelligent optimization algorithms, such as ant colony optimization (ACO) [1], particle swarm optimization (PSO) [2], grey wolf optimization (GWO) [3], Differential Evolution (DE) [4], artificial fish swarm algorithm (AFSA) [5]. Enlightened by the swarm foraging specific intelligent behavior, Karaboga proposed artificial bee colony algorithm [6] in 2005. ABC algorithm has many advantages compared with other intelligent optimization algorithm, such as less control parameters, great global optimization ability and easy of implementation. However, it has been shown that original ABC algorithm tends to suffer poor exploitation performance on complex problems [7-8]. To remedy this problem, many researchers have focused on the search rule as it controls the tradeoff between exploration and exploitation. Therefore, various new search strategies have been proposed in the literature [9-15].

This study presents an improved artificial bee colony algorithm. In our method, the solution search strategy applies the best solution, the second and third best solutions of the whole population to enhance the exploitation ability and accelerate the convergence speed. The new artificial bee colony algorithm is called IABC. The experimental results on sets of benchmark functions show that it could do well in solving complicated numerical optimization problems and perform better than other ABC-based algorithms.

The rest of this paper is organized as follows. The description about the ABC algorithm is in section 2. In Section 3, the IABC algorithm is proposed. The experimental results are provided and discussed in Section 4. Finally, Section 5 concludes the paper.

2. Basic artificial bee colony algorithm

The basic ABC algorithm is a population-based meta-heuristics algorithm that mimics the foraging behavior of honey bee swarms. In ABC algorithm, artificial bees are made up of employed bees, onlooker bees and scout bees. Employed bees are in charge of exploiting available food sources and collecting the information about these food sources. Then onlooker bees evaluate these information and make a decision by roulette to select better food sources and exploit around them. In ABC algorithm, a food source represents a possible solution of the optimization problem, the number of food sources and that of employed bees or onlooker bees is equal. If a food source is not enhanced for several runs, the employed bee becomes a scout bee and makes a random search for seeking a new food source.

The ABC algorithm consists a population of food sources with size SN. For the numerical optimization problem, each food source consists of a D-dimensional parameter vector, which encodes the candidate solution, i.e., $x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$, $i = 1, 2, \cdots, SN$.

At the beginning, the ABC algorithm generates an initial population of food sources randomly. The initial food sources are uniformly placed within the search space constrained by the predefined by lower and upper bounds $x_{\min}$ and $x_{\max}$. For parameter j in food source i, the initial value x_i^j is generated by

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

where $i = 1, 2, \cdots, SN$, $j = 1, 2, \cdots, D$. $rand(0,1)$ is a random value ranging in $[0,1]$.

After initialization, employed bees generate a candidate food source v_i by performing a local search around each food source ($i = 1, 2, \cdots, SN$) as follows:

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

where j is randomly selected dimension such that $j \in \{1, 2, \cdots, D\}$ and k is a randomly chosen food source such that $k \in \{1, 2, \cdots, SN\}$ and k is different from i. φ_{ij} is a random real number within the range $[-1,1]$. Once v_i is obtained, it is evaluated and compared with x_i. If the quality of v_i is better than x_i, v_i will replace x_i in the population. Otherwise, x_i will be remained in the population. In other words, a greedy selection is used between x_i and v_i.

Unlike the employed bees, the onlooker bees select a food source based on the probability value P_i as follows:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{3}$$

where fit_i denotes the fitness value of the solution x_i. Generally, the objective function value is directly used as fit_i in ABC algorithm.

After the onlooker bees choose the solutions, each one generates a new food source by Eq. (2) and the greedy selection is applied to select the better one from the new food source and the old one.

At last, when a food source cannot be improved through a predetermined number of cycles, i.e., parameter *limit*, then the corresponding employed bee becomes a scout bee. The scout bee generates a new food source according to Eq. (1).

3. Improved artificial bee colony algorithm

According to Eq. (2), it can be seen that the coefficient φ_{ij} is a random real number within the range $[-1,1]$ and x_k is a random food source. Therefore, Eq. (2) is good at exploration but poor at exploitation. In order to enhance the exploitation ability, we propose an improved search strategy for employed bees and onlooker bees as follows:

$$v_{ij} = \frac{x_{best1,j} + x_{best2,j} + x_{best3,j}}{3} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{4}$$

where x_{best1}, x_{best2} and x_{best3} is the best solution, the second and third best solution of the population. By incorporating the information of best solution, the second and third best solution into the search strategy for onlooker bees can increase the exploitation of ABC algorithm.

In basic ABC algorithm, the comparison of the new solution and the old solution is done by the fitness value, however, this method has shortcomings [7]. Thus, we use the objective function value for comparison and select better solution in this paper.

The main process of IABC algorithm is given below:

1: Set SN and the maximum number of functions evaluations $Max.FE$

2: Initialize the population of solutions $x_i (i = 1,2,\cdots,SN)$, and evaluate the population

3: While($FE < Max.FE$)

4: Produce new solution v_i for employed bees by using Eq.(4) and evaluate them

5: Apply greedy selection to determine the solution of employed bees

6: Calculate the probability values P_i for the solutions x_i by Eq.(3)

7: Produce new solution v_i for onlooker bees by using Eq.(4) and evaluate them

8: Apply greedy selection to determine the solution of onlooker bees

9: Determine the abandoned solution for the scout, if exist, and replace it with a new randomly produce solution x_i by Eq.(1)

10: Memorize the best solution achieved so far

11:End while ($FE = Max.FE$)

4. Experimental Study and Discussion

In order to investigate the effectiveness of the proposed IABC algorithm, 10 benchmark functions which have been widely used in many research works are employed in this paper. The definition of each function is presented in Table 1, where information of its mathematical formula, search range and the global optimum function values is included.

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
$f_4(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	[-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

Experiments on numerical function optimization were conducted for the proposed IABC, basic ABC algorithm and modified version ABC algorithms (GABC algorithm [9] and MABC algorithm [14]). For all the benchmark functions, the number of employed bees SN is set as 20, *limit* is $SN*D$, the number of maximum function evaluations is set to 150000. For each function, 30 independent runs are carried out, where the final results are the average values over these independent runs.

The mean and std (standard deviation) of searched best function values get by each algorithm are listed in Table 2. From the Table 2 results, it is found that the compared methods have most reliably found the minimum of the function f_3 and f_7, ABC algorithm has the best performance for the function f_5. For the rest of the functions, GABC algorithm, MABC algorithm and IABC algorithm perform better than basic ABC algorithm.

Table 2: Computational results on benchmark functions

Fun	Metric	ABC	GABC	MABC	IABC
f_1	Mean	4.93E-16	4.62E-16	9.43E-32	3.17E-56
	Std	7.98E-17	7.12E-17	6.67E-32	9.54E-56
f_2	Mean	1.31E-15	1.35E-15	2.40E-17	1.32E-37
	Std	1.54E-16	1.36E-16	9.02E-18	1.56E-37
f_3	Mean	8.37E-01	2.18E-01	1.02E+01	7.46E-03
	Std	4.72E-01	4.01E-02	1.49E+00	2.73E-03
f_4	Mean	0	0	0	0
	Std	0	0	0	0
f_5	Mean	4.32E-02	3.21E-01	6.11E-01	1.89E+00
	Std	4.51E-02	8.21E-01	4.55E-01	3.48E+00
f_6	Mean	4.85E-02	2.03E-02	3.71E-02	1.41E-02
	Std	1.29E-02	5.74E-03	8.53E-03	4.57E-03
f_7	Mean	0	0	0	0
	Std	0	0	0	0
f_8	Mean	1.78E-07	3.70E-17	0	0
	Std	8.89E-07	5.32E-17	0	0
f_9	Mean	3.55E-14	3.20E-14	4.13E-14	3.15E-14
	Std	3.62E-15	3.36E-15	2.17E-15	3.27E-15
f_{10}	Mean	3.18E-01	2.66E-01	2.95E-01	2.24E-01
	Std	5.19E-02	4.39E-02	3.17E-02	5.08E-02

5. Conclusion

To solve the problem of poor exploitation ability and slow convergence speed in basic artificial bee colony, an improved artificial bee colony algorithm was proposed in this study, called IABC algorithm. In the new algorithm, the employed bees and the onlooker bees utilize the best solution, the second and third best solutions of the whole population to enhance the exploitation ability and accelerate the convergence speed. The proposed IABC algorithm is compared with other ABCs on a set of 10 test functions. The comparison results demonstrate that the IABC algorithm was effective in getting solutions of excellent quality and smaller standard deviation for most of the experiments.

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]. S. Mirjalili, S M. Mirjalili, A. Lewis. Grey wolf optimizer. Advances in engineering Software, 69(3), 2014, 46-61.

[4]. R. Storn , K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 1997, 341-359.

[5]. X. L. Li, Z. J. Shao, J. X. Qian. An optimizing method based on autonomous animates: fish-swarm algorithm. System Engineering-Theory & Practice, 22(11), 2002, 32-38.

[6]. D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Turkey Kayserl: Erciyes University, 2005, 1-10.

[7]. A. Banharnsakun, T. Achalakul, B. Sirinaovakul. The best-so-far selection in artificial bee colony algorithm. Applied soft Computing, 11(2), 2011, 2888-2901.

[8]. W. F. Gao, S. Y. Liu. Improved artificial bee colony algorithm for global optimization. Information Processing Letters, 111(17), 2011, 687-697.

[9]. G. Zhu, S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation, 217(7), 2010, 3166-3173.

[10]. W. F. Gao, S. Y. Liu, L. L. Huang. A novel artificial bee colony algorithm with Powell's method. Applied soft Computing, 13(9), 2013, 3763-3775.

[11]. F. Kang, J. J. Li, Z. Y. Ma. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. Information Sciences, 181(16), 2011, 3508-3531.

[12]. B. Alatas. Chaotic bee colony algorithms for global numerical optimization. Expert System with Applications, 37(8), 2010, 5682-5687.

[13]. W. F. Gao, S. Y. Liu, L. 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. F. Gao, S. Y. 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.