



A New Artificial Bee Colony Algorithm for Solving Large-Scale Optimization Problems

Hui Wang^{1,2(✉)}, Wenjun Wang³, and Zhihua Cui⁴

¹ Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology,

Nanchang 330099, China

huiwang@whu.edu.cn

² School of Information Engineering, Nanchang Institute of Technology,

Nanchang 330099, China

³ School of Business Administration, Nanchang Institute of Technology,

Nanchang 330099, China

wangwenjun881@126.com

⁴ Complex System and Computational Intelligence Laboratory,

Taiyuan University of Science and Technology, Taiyuan 030024, China

zhihuacui@gmail.com

Abstract. Artificial bee colony (ABC) is an efficient global optimizer, which has been successfully used to solve various optimization problems. However, most of these problems are low dimensional. In this paper, we propose a new multi-population ABC (MPABC) algorithm to challenge large-scale global optimization problems. In MPABC, the population is divided into three sub-populations, and each subpopulation uses different search strategies. During the search, all subpopulations exchange their best search experiences to help accelerate the search. Experimental study is conducted on ten global optimization functions with dimensions 50, 100, and 200. Results show that MPABC is better than three other ABC variants on all dimensions.

Keywords: Artificial bee colony · Swarm intelligence · Multi-population Global optimization · Large-scale optimization

1 Introduction

Many real world problems can be formulated to optimization problems over continuous or discrete search space. Compared to traditional mathematical optimization techniques, bio-inspired optimization methods do not consider whether the optimization problems are continuous or differentiable. So, they can be easily used to solve complex optimization problems.

In the past decades, many bio-inspired optimization methods have been proposed, such as genetic algorithms (GAs) [1], simulated annealing (SA) [2], particle swarm optimization (PSO) [3], ant colony optimization (ACO) [4], artificial bee colony (ABC) [5], and others [6, 7]. Although these algorithms have been achieved success on many low-dimensional optimization problems, they suffer from the curse of

dimensionality. It means that their optimization performance deteriorates quickly with increasing of dimensions. To tackle this issue, some good algorithms were proposed in the literature [8–15].

ABC is one of the most popular optimization algorithm, which is inspired by the social behaviors of bees [16]. Since the introduction of ABC, it has been used to solve various optimization problems, but most of these problems are low-dimensional. To challenge large-scale global optimization problems, this paper proposes a new multi-population ABC (MPABC). Compared to the original ABC, MPABC employs three subpopulations, and each one use different search strategies. Ten benchmark optimization problems with dimensions 50, 100, and 200 are utilized in the experiments. Computational results show that MPABC is superior to three other ABC algorithms.

The rest of the paper is organized as follows. In Sect. 2, the original ABC is briefly described. Our approach MPABC is proposed in Sect. 3. Benchmark functions, results and discussions are presented in Sect. 4. Finally, this work is concluded in Sect. 5.

2 Artificial Bee Colony

In ABC, there are three different kinds of bees, employed, onlooker and scout. The number of employed bees is equal to the onlooker bees. The search of ABC is completed by different types of bees. Firstly, the employed bees search the neighborhood of each food source (solution) and find new better solutions. Secondly, the onlooker bees select some good solutions and search their neighborhoods to find better solutions. The scout bees randomly generate new solutions to replace the trapped ones.

For each solution X_i , an employed bee searches its neighborhood and find a new solution V_i [16].

$$v_{ij}(t) = x_{ij}(t) + \phi_{ij}(x_{ij}(t) - x_{kj}(t)), \quad (1)$$

where j is a random integer between 1 and D ; X_k is randomly selected from the population ($i \neq j$); t is the iteration index; ϕ_{ij} is a random value uniformly distributed with the range $[-1, 1]$. If V_i is better than its parent X_i , then replace X_i with V_i ; otherwise keep X_i unchangeable.

When all employed bees complete the search, the selection probability p_i for each food source X_i is calculated by [16]:

$$p_i = \frac{fit_i}{\sum_{i=1}^N fit_i}, \quad (2)$$

where fit_i is the fitness value of X_i . When a solution X_i is selected, an onlooker bee searches the neighborhood of X_i and obtain a new food source V_i according to Eq. (1). Like the employed bees, the onlooker bees also use the same method to compare V_i with X_i . If V_i is better than its parent X_i , then replace X_i with V_i ; otherwise keep X_i unchangeable.

If a solution X_i cannot be improved by employed or onlooker bees in *limit* iterations, it seems that X_i may be trapped into local minima. Then, a scout bee randomly generates a solution to replace X_i .

3 Proposed Approach

3.1 Multi-population Technique

In our previous study [17], we presented a multi-strategy ensemble ABC (MEABC), in which each food source is assigned a search strategy selected from a strategy pool. Results proved that ABC with two or more search strategies are better than that with a single strategy. Inspired by MEABC, we propose a new multi-population ABC (MPABC), which consists of three subpopulations, Subpop1, Subpop2, and Subpop3. Each subpopulation uses different search strategies to find new candidate solutions. In MPABC, Subpop1, Subpop2, and Subpop3 employ the original ABC, *gbest*-guided ABC (GABC) [18], and modified ABC (MABC) [19], respectively. Figure 1 shows the multi-population technique used in MPABC. As seen, all subpopulations share their best search experiences during the search.

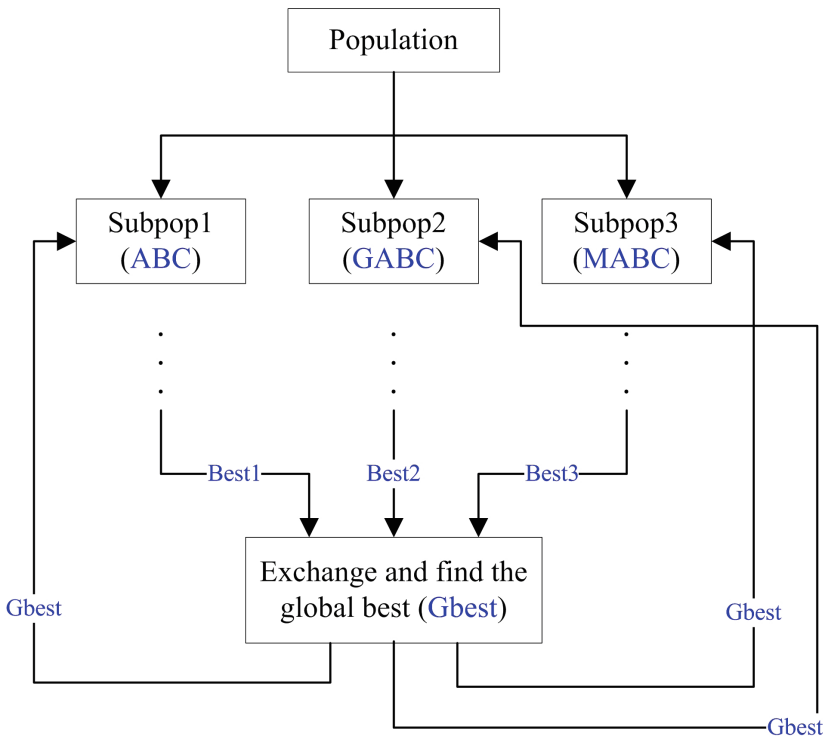


Fig. 1. The multi-population technique used in MPABC.

In the first subpopulation (Subpop1), MPABC uses the original ABC to execute the iteration and try to find new solutions. In the second subpopulation (Subpop2), MPABC employs GABC to execute the iteration and generate offspring. GABC and ABC are very similar, and they use the same framework and different search strategies. In GABC, a new search strategy incorporated with the best search experience is defined as follows.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(gbest_j - x_{ij}), \quad (3)$$

where φ_{ij} is a random number within $[0, C]$, and C is a constant value. $C = 1.5$ is suggested in [18].

In the third subpopulation (Subpop3), MPABC uses the MABC to execute the iteration and find new solutions. MABC is inspired by the differential evolution (DE) mutation, and it is defined by [19]:

$$v_{ij} = gbest_j + \phi_{ij}(x_{aj} - x_{bj}), \quad (4)$$

where X_a and X_b are two randomly selected solutions ($a \neq b \neq i$), and $gbest$ is the global best solution in the Subpop3.

3.2 Information Exchange

For multi-population technique, information exchange is an important operation, which can greatly affect the performance of algorithm. In MPABC, we use a new information exchange method. Assume that the population size is N . Each subpopulation consists of n food sources (solutions), and $n = N/3$. Every m fitness evaluations, all subpopulations exchange their best search experiences.

First, assume that the best solutions of Subpop1, Subpop2, and Subpop3 are Best1, Best2, and Best3, respectively. The best one Gbest is selected from Best1, Best2, and Best3 (please see Fig. 1). Then, we use Gbest to replace the $20\%*n$ solutions in each subpopulation. For Subpop1, we randomly selected $20\%*n$ solutions, and Gbest is assigned to these solutions. It is hopeful that Gbest can accelerate the search on large-scale optimization problems.

4 Experimental Study

4.1 Large-Scale Global Optimization Problems

There are ten large-scale global optimization problems used in the experiments. Problems F1-F6 were chosen from the CEC 2008 Special Session on large scale global optimization [20], and the rest problems F7-F10 were taken from the Special Issue of Soft Computing on large scale continuous optimization problems [21]. Table 1 present a brief description of the ten test problems. In the experiments, the problem dimension (D) is set to 50, 100, and 200.

Table 1. Ten large-scale global optimization problems used in the experiments.

Problems	Search range	Global optimum
Shifted Sphere Problem (F1)	[-100, 100]	-450
Shifted Schwefel's Problem 2.21 (F2)	[-100, 100]	-450
Shifted Rosenbrock's Function (F3)	[-100, 100]	390
Shifted Rastrigin's Function (F4)	[-5, 5]	-330
Shifted Griewank's Function (F5)	[-600, 600]	-180
Shifted Ackley's Function (F6)	[-32, 32]	-140
Shifted Schwefel's Problem 2.22 (F7)	[-10, 10]	0
Shifted Schwefel's Problem 1.2 (F8)	[-65.536, 65.536]	0
Shifted Extended f_{10} (F9)	[-100, 100]	0
Shifted Bohachevsky (F10)	[-15, 15]	0

4.2 Parameter Settings

This paper aims to use an improved ABC to challenge large-scale global optimization problems. Although several good bio-inspired optimization algorithms have been proposed to solve large-scale optimization problems, we only compare our approach MPABC with some ABC variants on the test suite. The compared algorithms are listed as follows.

- ABC;
- GABC [18];
- MABC [19];
- Our approach MPABC.

For all algorithms, the same parameter settings are used for common parameters. In ABC, GABC, MABC, and MPABC, the maximum number of fitness evaluations (*MaxFEs*) and population size (N), and *limit* are set to $5000 \cdot D$, 60, and 100, respectively. In GABC and MPABC, the parameter C is equal to 1.5 [18]. In MABC and MPABC, the parameter p is set to 0.7 [19]. The parameters m used in MPABC is set to 500 based on our empirical study. Because the population size N is 60, the size (n) of each subpopulation in MPABC is 20.

Each run stops when the maximum number of fitness evaluations is achieved. Throughout the experiments, the mean errors of the best solution found in the 25 runs are reported (For a solution X , the error value is calculated by $F(X) - F(X^o)$, where X^o is the global optimum of the problem).

4.3 Computational Results

Tables 2 presents the computational results of MAPBC, ABC, GABC, and MACB on problems with $D = 50$, where "Mean Error" indicates the mean error values between the best solution found so far and the global optimum. Compared to ABC, MPABC achieves better solutions on 8 problems. For the rest of 2 problems, both of them can converge to the global optimum. MPABC significantly improve the performance of

ABC on F4, F6, F7, and F9. GABC and MPABC find the same solutions on two problems F4 and F5. For the rest of 8 problems, MPABC is better than GABC. MABC and MPABC can find the global optimum on 4 problems F1, F4, F5, and F7, while MPABC outperforms MABC on the rest of 6 problems.

Table 2. Computation results for $D = 50$.

Problems	ABC	GABC	MABC	MPABC
	Mean error	Mean error	Mean error	Mean error
F1	0.00E+00	3.34E-25	0.00E+00	0.00E+00
F2	1.02E+02	6.77E+01	2.30E+01	3.21E+00
F3	4.50E+00	3.50E+00	5.35E+00	2.02E+00
F4	1.22E+00	0.00E+00	0.00E+00	0.00E+00
F5	5.73E-13	0.00E+00	0.00E+00	0.00E+00
F6	2.46E-06	1.17E-12	6.74E-14	5.86E-14
F7	7.01E-09	1.04E-13	0.00E+00	0.00E+00
F8	1.52E+04	1.34E+04	1.45E+04	8.08E+03
F9	2.77E+00	5.39E-02	8.48E-04	5.13E-05
F10	0.00E+00	3.31E-26	1.81E-36	0.00E+00

Table 3 gives the comparison results of MAPBC, ABC, GABC, and MABC on problems with $D = 100$. When the dimension increases to 100, ABC cannot converge to F1 and F10, and MPABC outperforms ABC on all problems. Especially for problems F1, F4, F5, F6, F7, F9, and F10, MPABC is much better than ABC. GABC performs better than MPABC on F3, but MPABC outperforms GABC on the rest of 9 problems. GABC falls into local minima on F1, F4, F5, F7 and F10, while our approach can find the global optimum. MABC is better than ABC and GABC. Both MABC and MPABC achieve the same results on F4 and F5. For the rest of 8 problems, MPABC outperforms MABC.

Table 4 presents the computational results of MAPBC, ABC, GABC, and MABC on problems with $D = 200$. As the dimension increases to 200, MPABC still converges to the global optimum on 4 problems F1, F4, F7, and F10. MABC can find the global optimum on only one problem F4. For ABC and GABC, they fall into local minima on all problems. MABC is slightly better than MPABC on F5 and F6, while MPABC outperforms MABC on 7 problems. MPABC achieves much better solutions than ABC and GABC on all problems.

In order to identify the significant differences between two algorithms, Wilcoxon test is conducted [22]. Tables 5, 6, and 7 present the p -values of applying Wilcoxon test among MPABC and other three ABC variants for $D = 50, 100$, and 200 , respectively. The p -values below 0.05 (the significant level) are shown in bold. As shown, MPABC is significantly better than ABC, GABC, and MABC for $D = 50$ and 100 . For $D = 200$, MPABC is only significantly better than ABC and GABC.

Table 3. Computation results for $D = 100$.

Problems	ABC	GABC	MABC	MPABC
	Mean error	Mean error	Mean error	Mean error
F1	5.20E-15	6.66E-24	4.77E-30	0.00E+00
F2	1.34E+02	1.20E+02	5.66E+01	1.35E+01
F3	1.47E+01	1.04E+01	4.05E+01	1.43E+01
F4	4.54E+00	1.03E-13	0.00E+00	0.00E+00
F5	2.54E-13	7.33E-16	0.00E+00	0.00E+00
F6	6.04E-06	3.64E-12	1.82E-13	1.37E-13
F7	2.07E-08	7.52E-13	6.96E-17	0.00E+00
F8	5.10E+04	5.88E+04	5.68E+04	3.14E+04
F9	7.46E+00	2.33E-01	7.65E-03	5.24E-05
F10	2.57E-14	8.88E-25	1.94E-30	0.00E+00

Table 4. Computation results for $D = 200$.

Problems	ABC	GABC	MABC	MPABC
	Mean error	Mean error	Mean error	Mean error
F1	8.93E-14	7.27E-23	5.76E-28	0.00E+00
F2	1.54E+02	1.51E+02	8.90E+01	3.58E+01
F3	1.57E+01	4.35E+01	2.97E+01	1.50E+01
F4	9.86E+00	2.86E-12	0.00E+00	0.00E+00
F5	7.67E-13	2.40E-15	1.11E-16	1.48E-16
F6	9.87E-06	1.08E-11	4.19E-13	5.43E-13
F7	8.65E-08	3.35E-12	3.75E-15	0.00E+00
F8	1.90E+05	2.13E+05	2.01E+05	1.14E+05
F9	1.75E+01	6.76E-01	3.86E-02	4.08E-04
F10	3.02E-13	1.29E-23	5.33E-29	0.00E+00

Table 5. Wilcoxon test between MPABC and the other three ABC variants for $D = 50$.

MPABC vs.	p -values
ABC	1.17E-02
GABC	1.17E-02
MABC	2.77E-02

Table 6. Wilcoxon test between MPABC and the other three ABC variants for $D = 100$.

MPABC vs.	p -values
ABC	5.06E-03
GABC	4.69E-02
MABC	1.17E-02

Table 7. Wilcoxon test between MPABC and the other three ABC variants for $D = 200$.

MPABC vs.	p -values
ABC	5.06E-03
GABC	5.06E-03
MABC	8.58E-02

5 Conclusions

In the past decade, many different ABC algorithms have been proposed to various optimization problems. However, most of these problem are low-dimensional. To challenge large-scale optimization problems, this paper presents an improved ABC variant (called MPABC), which employs a new multi-population. MPABC consists of three subpopulations, and they use ABC, GABC, and MABC to execute iterations and generate new solutions, respectively. During the search, each subpopulation exchange their best search experiences with others. To validate the performance of MPABC, ten large-scale global optimization problems with dimensions 50, 100, and 200 are utilized in the experiments.

Computational results show that MPABC is superior to ABC, GABC, and MABC on most test problems. As the dimension increases, the performance of ABC, GABC, and MABC is seriously affected, while MPABC still can achieve good solutions. It demonstrates that the proposed multi-population technique can effectively combine the advantages of ABC, GABC, and MABC during the search.

In this paper, we only test MPABC on $D = 50, 100, \text{ and } 200$. For problems with larger scale (such as $D = 500, 100, \text{ and } 2000$), we did not investigate the effectiveness of MPABC. Moreover, MPABC introduces two new parameters m and n . The first parameter determine the exchange gap. Different m may affect the convergence speed. The second parameter is the size of subpopulation. In MPABC, we assume that all subpopulations have the same size. For different sizes of subpopulations, we have not studied its effects. The above issues will be our research directions in the future work.

Acknowledgement. This work was supported by the Science and Technology Plan Project of Jiangxi Provincial Education Department (No. GJJ170994), the National Natural Science Foundation of China (No. 61663028), the Distinguished Young Talents Plan of Jiangxi Province (No. 20171BCB23075), the Natural Science Foundation of Jiangxi Province (No. 20171BAB-202035), and the Open Research Fund of Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing (No. 2016WICSIP015).

References

1. Schmitt, L.M.: Theory of genetic algorithms. *Theor. Comput. Sci.* **259**(1–2), 1–61 (2001)
2. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
3. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948 (1995)

4. Dorigo, M., Maniezzo, V., Coloni, A.: The ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **26**, 29–41 (1996)
5. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-TR06, Erciyes University, engineering Faculty, Computer Engineering Department (2005)
6. Wang, H., et al.: Firefly algorithm with neighborhood attraction. *Inf. Sci.* **382–383**, 374–387 (2017)
7. Cui, Z., Sun, B., Wang, G., Xue, Y., Chen, J.: A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems. *J. Parallel Distrib. Comput.* **103**, 42–52 (2017)
8. Wang, H., Wu, Z., Rahnamayan, S.: Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. *Soft Comput.* **15**(11), 2127–2140 (2011)
9. Brest, J., Maučec, M.S.: Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft Comput.* **15**(11), 2157–2174 (2011)
10. Long, W., Jiao, J., Liang, X., Tang, M.: Inspired grey wolf optimizer for solving large-scale function optimization problems. *Appl. Math. Model.* **60**, 112–126 (2018)
11. LaTorre, A., Muelas, S., Peña, J.M.: A comprehensive comparison of large scale global optimizers. *Inf. Sci.* **316**, 517–549 (2015)
12. Mahdavi, S., Shiri, M.E., Rahnamayan, S.: Metaheuristics in large-scale global continuous optimization: a survey. *Inf. Sci.* **295**, 407–428 (2015)
13. Mohapatra, P., Das, K.N., Roy, S.: A modified competitive swarm optimizer for large scale optimization problems. *Appl. Soft Comput.* **59**, 340–362 (2017)
14. Ali, A.F., Tawhid, M.A.: A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems. *Ain Shams Eng. J.* **8**(2), 191–206 (2017)
15. Hu, X.M., He, F.L., Chen, W.N., Zhang, J.: Cooperation coevolution with fast interdependency identification for large scale optimization. *Inf. Sci.* **381**, 142–160 (2017)
16. Akay, B., Karaboga, D.: A modified Artificial bee colony algorithm for real-parameter optimization. *Inf. Sci.* **192**, 120–142 (2012)
17. Wang, H., Wu, Z.J., Rahnamayan, S., Sun, H., Liu, Y., Pan, J.S.: Multi-strategy ensemble artificial bee colony algorithm. *Inf. Sci.* **279**, 587–603 (2014)
18. Zhu, G., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl. Math. Comput.* **217**, 3166–3173 (2010)
19. Gao, W., Liu, S.: A modified artificial bee colony algorithm. *Comput. Oper. Res.* **39**, 687–697 (2012)
20. Tang, K., et al.: Benchmark functions for the CEC’2008 special session and competition on large scale global optimization. Nature Inspired Computation and Applications Laboratory, USTC, China (2007)
21. Herrera, F., Lozano, M., Molina, D.: Test suite for the special issue of Soft Computing on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems. Technical report, University of Granada, Spain (2010)
22. Wang, H., Rahnamayan, S., Sun, H., Omran, M.G.: Gaussian bare-bones differential evolution. *IEEE Trans. Cybern.* **43**(2), 634–647 (2013)