Artificial bee colony with multiple search strategies and a new updating mechanism

Xin Li, Kai Li, Tao Zeng, Tingyu Ye, Luqi Zhang, Hui Wang*

School of Information Engineering,
Nanchang Institute of Technology,
Nanchang 330099, China
E-mail: nclx2019@163.com
E-mail: Aha15380251715@gmail.com
E-mail: nitztao@163.com
E-mail: yty01122@163.com
E-mail: nczlq2020@126.com
E-mail: huiwang@whu.edu.cn
*Corresponding author

Abstract: The imbalance of exploration and exploitation is a weakness in artificial bee colony (ABC) algorithm. To overcome this deficiency, this paper presents an improved ABC (namely IABC) by employing multiple search strategies and a novel updating method. Firstly, a concept of marginal group is introduced to construct an exploration search strategy. Then, an exploitation search strategy is designed utilizing some excellent solutions. Thirdly, the probabilistic selection strategy is modified on this basis of some elite solutions. In the experiments, 22 benchmark problems were utilized to prove the effectiveness of IABC. Test results indicate that IABC achieves stronger optimisation capabilities than the other five ABCs.

Keywords: Artificial bee colony; marginal group; multiple search strategies; selection mechanism.

Reference to this paper should be made as follows: Li, X., Li, K., Zeng, T., Ye, T. Y., Zhang, L. Q. and Wang, H. (2021) 'Artificial bee colony with multiple search strategies and a new updating mechanism', *International Journal of Computing Science and Mathematics*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Xin Li is an undergraduate at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes swarm intelligence and global optimization.

Kai Li is an undergraduate at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes swarm intelligence and global optimization.

Tao Zeng is a graduate student at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes intelligent optimization and application.

Tingyu Ye is a graduate student at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes intelligent optimization and application.

Luqi Zhang is a graduate student at the School of Information Engineering, Nanchang Institute of Technology, Nanchang, China. His research interest includes intelligent optimization and application.

Hui Wang is a professor at the School of Information Engineering, Nanchang Institute of Technology, China. His research interests include evolutionary optimization, swarm intelligence, and large-scale optimization. He has published more than 50 research papers in these fields.

1 Introduction

Swarm intelligence algorithms (IOAs) can show promising performance in addressing complex

Copyright © 201X Inderscience Enterprises Ltd.

optimization problems and have attracted much attention. Some representative IOAs include: ant colony algorithm (ACO) (Zhang et al., 2000), particle swarm algorithm (PSO) (Liu et al., 2005) and artificial bee colony (ABC) (Karaboga, 2011).

Of the algorithms mentioned above, ABC has a strong global search capability as well as few control parameters. However, it has some disadvantages in resolving some complex optimization problems, such as unbalanced exploration and exploitation capabilities and the tendency to fall into local optimization. To combat these disadvantages, many scholars have proposed various effective methods (Xiao et al., 2019; Gao et al., 2013; Li et al., 2021; Ye et al., 2021; Zeng et al., 2021; Yurtkuran et al., 2015).

To counterbalance the exploration and exploitation of ABC, an improved ABC algorithm with multiple search strategies and a new update mechanism (namely IABC) is proposed. In IABC, the following three main improvements: 1) a concept of marginal group is introduced to construct an exploration search strategy; 2) an exploitation search strategy is designed utilizing some excellent solutions; and 3) a new selection method is suggested in the onlooker bee phase as well as a modification of the corresponding search strategy at this phase. To prove that IABC performs, 22 benchmark problems were utilized. Test results indicate that IABC has stronger optimisation capabilities than the other five ABC algorithms.

Arrangements for the remainder of this work are set out below. Section 2 depicted the ABC. Section 3 explains the suggested IABC algorithm. Section 4 gives the results and analysis of the experiments. Lastly, a conclusion to this work is provided in section 5.

2 Artificial bee colony

ABC is a novel proposed optimisation technique for simulating the feeding behaviour of bee colonies. The whole algorithm is structured into several main phases as below.

Initialization At this phase, SN food sources (solutions) are randomly derived by Eq. (1).

$$X_{i,j} = X_{low,j} + \varphi \cdot (X_{up,j} - X_{low,j}), \tag{1}$$

where $i = \{1, 2, \ldots, SN\}$, $j = \{1, 2, \ldots, D\}$, D is the problem dimension, X_i is the i-th solution, $\varphi \in [0, 1]$ is a random value, and $[X_{low}, X_{up}]$ is the search boundary.

Employed bee phase A new solution V_i is found in the neighborhood of the current solution X_i by Eq. (2).

$$V_{i,j} = X_{i,j} + \phi_{i,j} \cdot (X_{i,j} - X_{k,j}), \tag{2}$$

where $\phi_{i,j} \in [0, 1]$ and $k \in \{1, 2, \ldots, SN\}$ are random numbers, $k \neq i$. If V_i offers a superior fitness value over X_i, X_i to be replaced by V_i. The fitness value for each solution is computed by Eq. (3).

$$fit(X_i) = \begin{cases} \frac{1}{f(X_i)+1}, & \text{if } f(X_i) \geq 0 \\ 1 + |f(X_i)|, & \text{otherwise} \end{cases}, \tag{3}$$

where $f(X_i)$ and $fit(X_i)$ are the function value and fitness value for X_i, respectively.

Onlooker bee phase Some good food sources (solutions) are chosen for onward exploitation according to the roulette selection mechanism. The search strategy and update method for food sources (solutions) in this phase are identical to the employed bee phase. Each solution has a probability p_i of being selected defined as

$$p_i = \frac{fit(X_i)}{\sum_{i=1}^{SN} fit(X_i)} . \tag{4}$$

According to the probability p_i, X_i is chosen for further search by Eq. (2).

Scout bee phase A solution is dropped when it has not changed after *limit* times. A new solution is initialized by Eq. (1) and is used instead of the discarded solution.

3 The proposed algorithm IABC

3.1 Marginal group

In ABC, there are SN solutions in the population. The population center $C = (C_1, C_2, \cdots, C_D)$ can be defined by

$$C_j = \frac{1}{SN} \cdot \sum_{i=1}^{SN} X_{i,j}. \tag{5}$$

where C_j denotes the j-th dimension for the population center C.

Within this paper, a new concept of marginal group is firstly defined. Assume that $M = \{M_1, M_2, \ldots, M_{N_1}\}$ is the marginal group, where $N_1 \in [0, SN]$ is the size of the marginal group. Each member in M is far from the population center C. Initially, N_1 is equal to 0. In the course of the search process, the marginal group is updated as below.

1. For each solution X_i in the population, the Euclidean distance Dis_i between X_i and the population center C is calculated as below.

$$Dis_i = \sqrt{\sum_{j=1}^{D} (C_j - X_{i,j})^2}, \tag{6}$$

where $i = 1, 2, \cdots, SN$.

2. The maximum distance Dis_m ($m \in [1, SN]$) is chosen from the set $\{Dis_1, Dis_2, \ldots, Dis_{SN}\}$,

where $Dis_m > Dis_i$ and $i = 1, 2, \ldots, SN$. Then, the corresponding X_m is regarded as the farthest solution from the population center C. When X_m is not in the marginal group, it will be stored in the marginal group. If $N_1 < SN$, X_m is directly added to the marginal group. If the marginal group is full ($N_1 == SN$), a solution is randomly chosen and removed from the marginal group, and X_m is stored into the marginal group.

3.2 Multiple search strategies

Aiming at the imbalance between exploration and exploitation, two new search strategies for the employed bee phase are proposed. One of these search strategies prefers exploration, and it is called the exploration strategy. The other strategy is biased toward exploitation, and it is called the exploitation strategy.

In the exploration strategy, the difference between a solution M_g randomly selected from the marginal group and the population center C is used as the search step for the bees. Since M_g is far from C, the search step is large. Then, the search range of the bees can be expanded. The exploration strategy can be defined as below.

$$V_{i,j} = X_{i,j} + \phi_{i,j} \cdot (M_{g,j} - C_j), \tag{7}$$

where $g = \{1, 2, \ldots N_1\}$ is a random integer.

In the exploitation strategy, the current global best solution *Gbest* is introduced to quickly improve the quality of solutions. However, a single-minded search using *Gbest* is likely to result in local minima. To avoid this problem, the search information of some other excellent solutions is introduced in the search strategy, such as the previous global best solution *LGbest* and elite group *Elite*. *LGbest* can provide the population with excellent historical location information. The solutions in the current population are all ranked from best to worst according to their function values. The top N_2 solutions are selected to form the *Elite*, where $N_2 = w \cdot SN$ is a predefined parameter and $w \in (0, 1]$. As seen, the *Elite* can provide some excellent local information. Then, the exploitation strategy is defined as below.

$$V_{i,j} = \beta + \phi_{i,j} \cdot (\beta - X_{k,j}), \tag{8}$$

$$\beta = \frac{1}{3} \cdot (Gbest_j + LGbest_j + Elite_{g,j}), \tag{9}$$

where $g = \{1, 2, \ldots N_2\}$ is a random number, and $Elite_g$ is the g-th solution in the elite group.

At the beginning, all employed bees in the population use the exploration strategy. During the iterative process, the search strategy with the largest search *limit* is dynamically updated with a probability p. The updating method can be described as below.

$$\begin{cases} \text{Use another strategy,} & \text{if } r > p \\ \text{Keep the current strategy,} & \text{otherwise} \end{cases}, \tag{10}$$

$$p = \frac{1}{2} \cdot (s + q), \tag{11}$$

$$s = \frac{N_{success}}{SN}, \tag{12}$$

$$q = \frac{FEs}{MaxFEs}, \tag{13}$$

where $r \in [0, 1]$ is a random number, $p \in [0, 1]$ represents the probability of updating the search strategy. $s \in [0, 1]$ is the success rate of bees, and $N_{success} \in [0, SN]$ is the number of successful searches in the previous generation. $q \in [0, 1]$ is a control factor, which can balance the effects of s on p. FEs and $MaxFEs$ are the present and the maximum number of function evaluations, respectively. If $r > p$, another search strategy is selected to replace the current one; otherwise, keep the current search strategy unchangeable.

3.3 New selection method

In this paper, the onlooker bees do not use the probability selection method to select good solutions for further search. To assign more computing resources towards some of the better solutions, only the global best solution and solutions in the elite group are selected by the bees. Moreover, the search equation of the onlooker bees is modified as below.

$$V_{i,j} = X_{i,j} + \phi_{i,j} \cdot (X_{i,j} - Elite_{g,j}), \tag{14}$$

where $g = \{1, 2, \ldots, N_2\}$ are random numbers, $Elite_g$ is taken from the elite group, and $Elite_g \neq X_i$

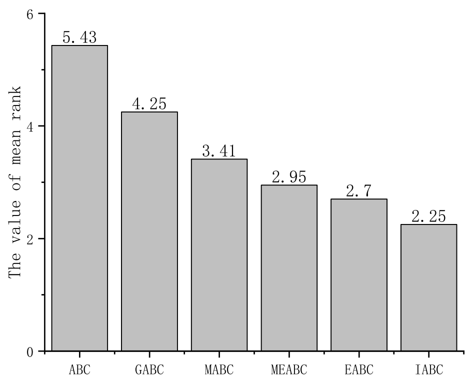


Figure 1 Mean rank values obtained for six ABCs (D=30).

4 Experimental study

To assess the performance of IABC, 22 benchmark problems were used. The details of these problems can be found in (Xiao et al., 2021). In this experiment, IABC is compared with EABC (Gao et al., 2014), ABC (Karaboga, 2005), MABC (Gao et al., 2012), GABC (Zhu et al., 2010) and MEABC (Wang et al.,

Table 1 Results of the six ABCs ($D = 30$).

Problems	Average					
	ABC	GABC	MABC	MEABC	EABC	IABC
f_1	9.36E-16	4.43E-16	1.70E-37	8.53E-44	3.73E-64	**2.60E-76**
f_2	1.54E-09	4.36E-16	9.17E-28	7.37E-42	2.47E-60	**4.06E-70**
f_3	6.47E-16	4.99E-16	2.53E-37	1.15E-43	3.42E-65	**5.43E-77**
f_4	3.33E-17	2.09E-17	5.45E-68	1.63E-86	**1.81E-110**	2.37E-80
f_5	1.42E-10	1.36E-15	2.51E-20	3.94E-23	7.14E-34	**4.33E-39**
f_6	1.23E+01	2.90E+00	1.58E+01	3.30E+00	8.26E-01	**6.55E-01**
f_7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_8	7.14E-20	2.59E-20	**7.18E-66**	**7.18E-66**	**7.18E-66**	**7.18E-66**
f_9	7.12E-02	2.57E-02	3.02E-01	3.21E-02	1.35E-02	**1.31E-02**
f_{10}	8.76E-01	3.49E+00	4.71E-01	1.70E+00	1.60E+01	**2.12E-01**
f_{11}	7.11E-15	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{12}	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{13}	5.43E-16	2.11E-16	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{14}	-1.25E+04	**-1.26E+04**	**-1.26E+04**	**-1.26E+04**	**-1.26E+04**	**-1.26E+04**
f_{15}	1.42E-04	1.41E-13	8.13E-14	3.24E-14	**2.89E-14**	**2.89E-14**
f_{16}	7.72E-16	5.10E-16	**1.57E-32**	**1.57E-32**	**1.57E-32**	**1.57E-32**
f_{17}	7.77E-16	3.41E-16	**1.35E-32**	**1.35E-32**	**1.35E-32**	**1.35E-32**
f_{18}	1.86E-06	2.97E-07	4.63E-19	**2.10E-22**	1.22E-16	1.20E-17
f_{19}	2.81E-13	4.99E-16	**1.35E-31**	**1.35E-31**	**1.35E-31**	**1.35E-31**
f_{20}	3.41E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{21}	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**
f_{22}	-2.93E+01	**-2.96E+01**	**-2.96E+01**	**-2.96E+01**	**-2.96E+01**	**-2.96E+01**
$\boldsymbol{b/s/w}$	19/3/0	15/7/0	9/12/1	8/12/2	8/13/1	-/-/-

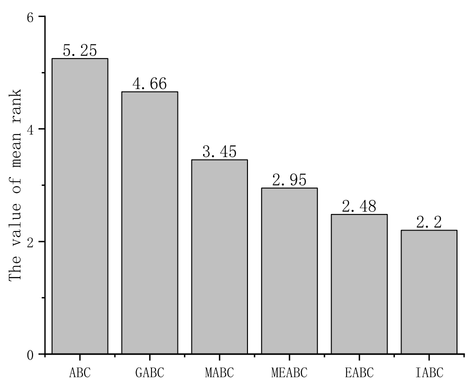


Figure 2 Mean rank values obtained for six ABCs ($D = 100$).

2014). The algorithms were all run on benchmarks of $D = 30$ and 100.

In our approach IABC, the parameter $SN = 50$, the stopping condition $MaxFEs = 5000 \cdot D$, $limit = SN \cdot D$ and $w = 0.1$ are utilized. For each problem, all algorithms run 30 times separately. For other parameter settings in EABC, MABC, GABC and MEABC, please refer to their corresponding literature (Gao et al., 2014, 2012; Zhu et al., 2010; Wang et al., 2014).

Table 1 displays the outcomes of six algorithms with $D = 30$. In the final line, "b/s/w" denotes that IABC is better, is similar to, and is worse than the other five algorithm for the correspondent number of problems, respectively. For instance, IABC outperforms ABC on 19 problems and both of them perform similarly on the other three problems. IABC performed better on 15 problems compared to the GABC, and they obtained similar results on the remaining

Table 2 Results of the six ABCs ($D = 100$).

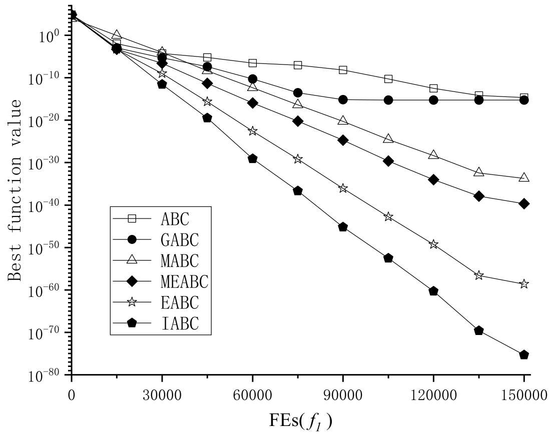
Problems	Average					
	ABC	GABC	MABC	MEABC	EABC	IABC
f_1	4.10E-15	2.51E-15	3.39E-33	2.18E-39	1.51E-59	**3.20E-72**
f_2	2.18E-09	1.86E-15	2.96E-28	5.17E-38	4.17E-55	**1.54E-70**
f_3	2.99E-15	2.10E-15	1.43E-33	3.12E-40	7.33E-60	**8.71E-73**
f_4	4.80E-17	2.61E-17	8.94E-66	6.97E-86	**8.96E-110**	1.80E-78
f_5	2.27E-09	5.59E-15	1.21E-17	2.60E-21	1.87E-31	**1.75E-36**
f_6	5.49E+01	3.76E+01	3.39E+01	3.61E+01	2.24E+01	**1.37E+01**
f_7	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_8	5.80E-33	1.40E-31	**7.12E-218**	**7.12E-218**	**7.12E-218**	**7.12E-218**
f_9	1.96E-01	1.59E-01	1.32E-01	1.69E-01	8.83E-02	**6.36E-02**
f_{10}	8.34E-01	1.42E+00	1.49E+01	2.90E-01	2.66E-01	**1.43E-01**
f_{11}	1.24E-14	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{12}	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{13}	1.44E-15	1.44E-15	1.19E-15	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{14}	-4.08E+04	-4.17E+04	**-4.19E+04**	**-4.19E+04**	**-4.19E+04**	**-4.19E+04**
f_{15}	2.91E-05	1.11E-09	4.05E-13	**1.14E-13**	**1.14E-13**	**1.14E-13**
f_{16}	3.12E-15	2.31E-15	**4.71E-33**	**4.71E-33**	**4.71E-33**	**4.71E-33**
f_{17}	7.87E-15	3.31E-15	**1.35E-32**	**1.35E-32**	**1.35E-32**	**1.35E-32**
f_{18}	4.43E-04	1.81E-05	1.34E-07	**8.43E-20**	2.22E-16	1.63E-16
f_{19}	3.16E-14	2.06E-15	**1.35E-31**	**1.35E-31**	**1.35E-31**	**1.35E-31**
f_{20}	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{21}	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**	**-7.83E+01**
f_{22}	-9.80E+01	-9.85E+01	**-9.96E+01**	**-9.96E+01**	**-9.96E+01**	**-9.96E+01**
$\boldsymbol{b/s/w}$	18/4/0	17/5/0	11/11/0	7/13/2	8/13/1	-/-/-

seven problems. IABC achieved superior outcomes than MABC on nine problems, while MABC surpasses IABC on only one problem f_{18}. They performed similarly on the remaining 12 problems. IABC is superior to MEABC on eight problems, and they are similar on 12 problems. For f_4 and f_{18}, MEABC outperforms IABC. EABC outperformed IABC on only f_4. For the remaining 21 problems, IABC is no worse than EABC. From the overall experimental results, IABC performs poorly on only two problems f_4 and f_{18}, while it is better or similar to the other five ABCs on the remaining 20 problems.

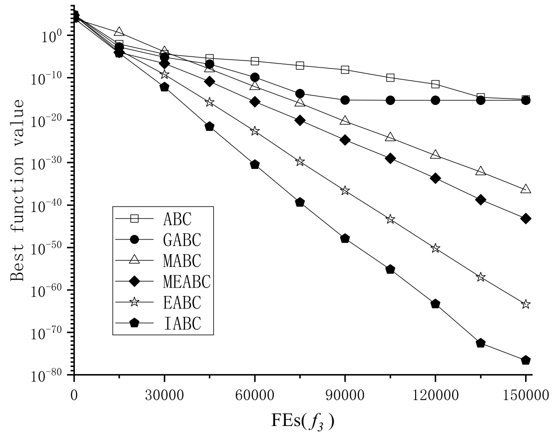
Table 2 presents the outcomes of the six ABCs for $D = 100$. As shown, all algorithms show similar performance on f_7, f_{12}, and f_{22}. IABC is not worse than ABC, GABC and MABC on all problems. In comparison to MEABC, IABC obtained worse results on f_4 and f_{18}. For the comparison results of IABC and EABC with $D = 30$, their comparison results are not changed for $D = 100$. As can be noticed from the experimental results that the increase of dimension D has no significant impact on the performance of IABC.

To further evaluate the overall capability of the six ABCs, Friedman test (Xiao et al., 2021) are used. Fig. 1 and Fig. 2 show the mean rank values for the six ABCs. As can be seen, IABC achieves the optimum performance for the six ABCs at $D = 30$ and 100.

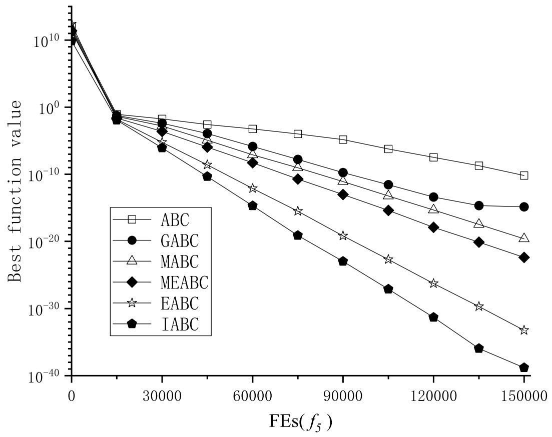
Fig. 3 shows the convergence curves of these ABCs on f_1, f_3, f_5, f_6, f_9, and f_{15} for $D = 30$. From the graphs, IABC converges fastest on f_1, f_3, f_5, and f_9. For f_6 and f_{15}, the convergence rate of IABC is not the quickest in the early stage. With increasing numbers of iterations, the convergence speed of IABC progressively exceeds that of the other five algorithms.



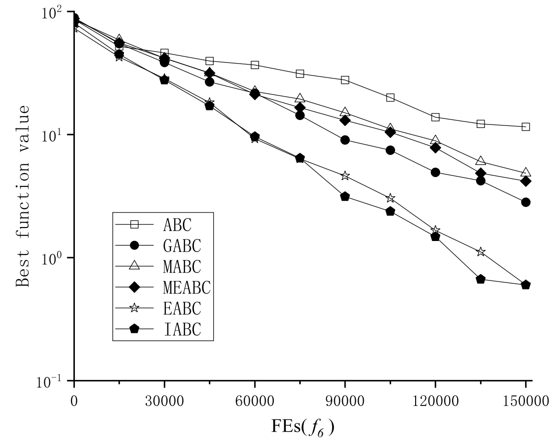
(a) Sphere (f_1)



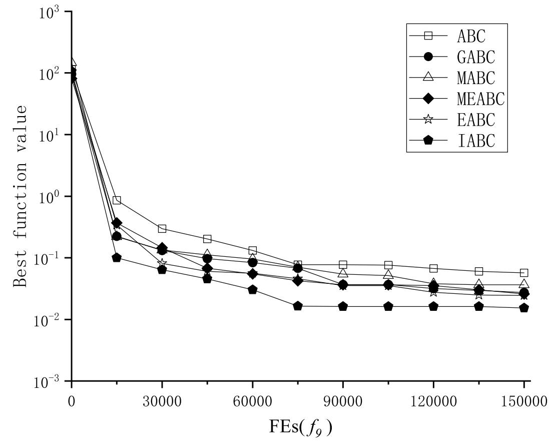
(b) SumSquare (f_3)



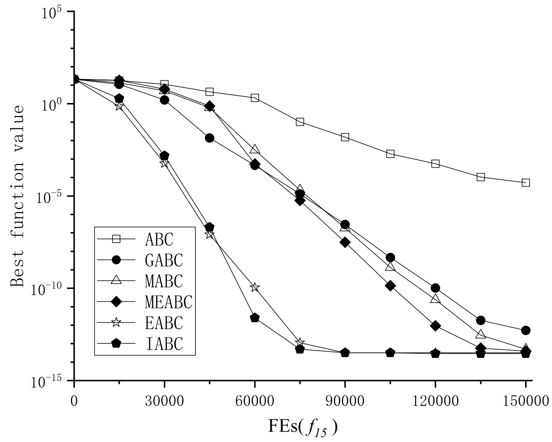
(c) Schwefel 2.22 (f_5)



(d) Schwefel 2.21 (f_6)



(e) Quartic (f_9)



(f) Ackley (f_{15})

Figure 3 Convergence curves of six ABCs on f_1, f_3, f_5, f_6, f_9, and f_{15} for $D = 30$.

5 Conclusion

This paper proposes an improved ABC (namely IABC) to balance exploitation and exploration. In IABC, there are three important modifications: 1) the marginal group is used to enhance the jumping ability of bees; 2) two new search strategies and a new updating mechanism are designed; and 3) the onlooker bees use a modified search equation and a novel selection strategy. To prove the effectiveness of IABC, 22 benchmark problems were tested. Results indicate IABC performs poorly on only two problems, while it outperforms or is similar to other five ABCs on the remaining 20 problems. From the perspective of convergence characteristics, IABC converges faster on most problems than the other five ABCs.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 62166027), and Jiangxi Provincial Natural Science Foundation (Nos. 20212ACB212004 and 20212BAB202023).

References

Zhang, J. , Gao, Q. and Xinhe, X. U. . (2000). A self-adaptive ant colony algorithm. Control *Theory and Applications*, 17(1).

Liu, B. , Wang, L. , Jin, Y. H. , Fang, T. and Huang, D. X. . (2005). Improved particle swarm optimization combined with chaos. *Chaos Solitons and Fractals*, 25(5), 1261-1271.

Karaboga, D. and Akay, B. . (2011). A modified artificial bee colony (abc) algorithm for constrained optimization problems. *Applied Soft Computing Journal*, 11(3), 3021-3031.

Xiao, S., Wang, H., Wang W. and Tan, et al. (2019). An improved artificial bee colony algorithm based on elite strategy and dimension learning. *Mathematics.*

Gao, W. , Liu, S. and Huang, L. . (2013). A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. *IEEE Transactions on Cybernetics*, 43(3), 1011-1024.

Li, K., Wang, H., Wang, W., Wang, F. and Cui, Z. (2021). Improving Artificial Bee Colony Algorithm Using Modified Nearest Neighbor Sequence. *Journal of King Saud University - Computer and Information Sciences.*

Ye, T. , Zeng, T. , Zhang, L. , Xu, M. and Hu, M. . (2021). Artificial Bee Colony Algorithm with an Adaptive Search Manner. *In International Conference on Neural Computing for Advanced Applications*, 486-497.

Zeng, T. , Ye, T. , Zhang, L. , Xu, M. and Hu, M. . (2021). Population Diversity Guided Dimension Perturbation for Artificial Bee Colony Algorithm.*In International Conference on Neural Computing for Advanced Applications*, 473-485.

Yurtkuran, A. and Emel, E. . (2015). An adaptive artificial bee colony algorithm for global optimization. *Applied Mathematics and Computation*, 271, 1004-1023.

Wang, H. , Wu, Z. , Rahnamayan, S. , Sun, H. , Liu, Y. and Pan, J. S. . (2014). Multi-strategy ensemble artificial bee colony algorithm. *INFORMATION SCIENCES*, 279, 587-603.

Karaboga, D. . (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report-TR06, Erciyes University, Engineering Faculty, Computer engineering Department.*

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

Gao, W. F. and Liu, S. Y. . (2012). A modified artificial bee colony algorithm. *Comput. Oper. Res*, 39, 687-697.

Gao, W. F. , Liu, S. Y. and Huang, L. L. . (2014). Enhancing artificial bee colony algorithm using more information-based search equations. *Information ences*, 270, 112 ÍC 133.

Xiao, S. , Wang, H. , Wang, W. , Huang, Z. and Xu , M. . (2021). Artificial bee colony algorithm based on adaptive neighborhood search and gaussian perturbation. *Applied Soft Computing*, 100(3), 106955.