



Artificial Bee Colony Algorithm Based on New Search Strategy

Minyang Xu^{1,2}, Hui Wang^{1,2(✉)}, Songyi Xiao^{1,2}, and Wenjun Wang³

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

Abstract. Artificial bee colony (ABC) is a popular swarm intelligence algorithm. In ABC, a food source is considered as a feasible solution. Bees flying in the sky and searching food sources is converted into an optimization process. In contrast with other swarm intelligence algorithms, ABC has fewer parameters and stronger search ability. Though ABC excels at exploration, it does not perform well in exploitation. This paper proposes an improved ABC algorithm based on a new search strategy (called NSSABC), in which some current global best solutions are preserved and they are used to guide the search. Experiment was performed on some classical problems and results shows the proposed strategy greatly improves the optimization ability of ABC.

Keywords: Artificial bee colony algorithm · External archive · Global best solution · New search method

1 Introduction

Swarm intelligence algorithms have some advantages in handling complex and difficult optimization problems [1–3]. So, it has received more and more attention in optimization community. At present, the main swarm intelligence algorithms include ACO, PSO and ABC. Compared with PSO and ACO, ABC has fewer parameters. Some other studies proved that ABC has certain advantages [3–8].

However, ABC also has certain disadvantages. Some studies showed that the randomness of the neighborhood selection method is too strong [4–8]. So, ABC exhibits good exploration and poor exploitation. To conquer this shortcoming, there were some improved methods for search strategy design [9–17]. For example, an enhanced ABC based on global best solution was proposed in [9]. In [10], Wang et al. utilized an external archive to guide ABC. In [11], different guidance information was used to improve the solution. Banharnsakun et al. [12] used a best-so-far ABC to effectively solve the JSSP problem. In [13], ABC was parallelized [13]. Tuba et al. [14]

introduced adaptive steering adjustments in ABC. In [15], inertia weights and acceleration factors were used to strengthen the search equation. Bi and Wang [16] designed a fast mutation-based ABC. In [17], DE was utilized to strengthen the search equation. It can be seen from the above literature that most of improvements on ABC aim to improve their search equations.

To further strengthen the performance, an improved ABC based on external archive is presented. The new method is motivated by GABC [9] and JADE [18]. In NSSABC, we use an external archive [10] to preserve the global best solution in the search equation of GABC. Those best solutions contained in the external archive are updated as the number of iterations increases. Simulation test experiments show NSSABC algorithm is superior to the standard ABC, GABC and IABC on most test functions.

The remainder of this paper is organized as follows. The descriptions of ABC are briefly given in Sect. 2. The proposed NSSABC is described in Sect. 3. Simulation experiments are conducted in Sect. 4. Finally, this work is concluded in Sect. 5.

2 Artificial Bee Colony

In ABC, there are four main processes: initialization process, employ bee process, onlooker bee process, and scout bee process. At the initialization stage, ABC randomly generates SN solutions (food sources) by the following formula.

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

where $i = 1, 2, \dots, SN$, $j = 1, 2, \dots, D$, SN is the population size, D is the dimension size, $rand(0, 1)$ randomly generated in $[0, 1]$ and $[x_{j,\min}, x_{j,\max}]$ is the constraint of search range.

In the process of employing bee, they have their own food sources. Each employ bee finds other food sources around its own food source by the following formula.

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{2}$$

where $i = 1, 2, \dots, SN$, v_i is a new solution, x_k is a randomly chosen solution ($k \neq i$), and $\phi_{i,j} \in [-1, 1]$ is a random weight.

In the process of onlooker bee, they select the food source according to the selection formula as follows.

$$p_i = \frac{f_i}{\sum_{i=1}^{SN} f_i} \tag{3}$$

where p_i is the selection probability, and f_i is the fitness value. It is apparently that the probability and the fitness value are positively correlated. It means that a better solution has a larger selection probability.

In the process of scout bee, if the food source could not be further updated within some iterations, it is supposed to be discarded. Assuming that the discarded solution is x_i , the scout bee randomly produces a new one by formula (1) instead of x_i .

3 ABC Based on New Search Strategy

Global exploration capabilities and local exploitation capabilities are two very important aspects for assessing the optimization ability of a swarm intelligence algorithm. So, balancing the two capabilities has influences on the performance of ABC.

For example, in GABC, the global best solution is used as a directive solution. Compared with the standard ABC, GABC further enhances the exploitation capabilities. However, the gbest may cause the entire swarm to a wrong search direction. It is pointed out in [7] that the search formula in GABC may be in the opposite direction, which easily leads to the oscillation during the search.

In [10], an external archive in IABC was used to lead the search, in which a concept of external archive makes full use of the global optimal solution. However, IABC causes all solutions converging toward the global best solution at different stages. It may result in the loss of swarm diversity.

To solve the above problems, a new ABC based on external archive is designed, which combines IABC and GABC. In NSSABC, we add the external archive of IABC to the GABC search strategy to replace the guidance of the gbest in GABC. The external archive focuses on preserving different global best solutions. With the growth of iterations, the external archive is updated. Compared with GABC, NSSABC uses some good solutions (not the best one) to lead the search. It may correct the wrong search direction and avoid local minima. Unlike IABC, NSSABC does not use the current gbest to lead the search. It reduces the degree of convergence of all solutions to the gbest, and the swarm diversity increases. In summary, NSSABC can maintain strong exploration and exploitation capabilities.

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) + \psi_{i,j}(x_{Arc,j} - x_{i,j}) \quad (4)$$

where x_{Arc} is randomly chosen from the external archive, and x_k is randomly selected from the swarm ($k \neq i$). $\psi_{i,j} \in [0, 1.5]$ and $\phi_{i,j} \in [-1, 1]$ are two random numbers.

Initially, an external archive A is created, and m is used as the threshold size of the archive A . After each generation, the archive A is updated. If the gbest is improved, then gbest will be preserved in A and the size increases 1 [10]. If the current archive is greater than m , we randomly delete a solution in the archive A .

Compared with the standard ABC, NSSABC only modifies the search strategy. Therefore, they have the same time complexity.

Table 1. Benchmark functions

Functions	Search range	Global optimum
Sphere (f_1)	[-100,100]	0
Schwefel 2.22 (f_2)	[-10,10]	0
Schwefel 1.2 (f_3)	[-100,100]	0
Schwefel 2.21 (f_4)	[-100,100]	0
Rosenbrock (f_5)	[-30,30]	0
Step (f_6)	[-100,100]	0
Quartic with noise (f_7)	[-1.28,1.28]	0
Schwefel 2.26 (f_8)	[-500,500]	-12569.5
Rastrigin (f_9)	[-5.12,5.12]	0
Ackley (f_{10})	[-32,32]	0
Griewank (f_{11})	[-600,600]	0
Penalized (f_{12})	[-50,50]	0

4 Experimental Study

To assess the optimization ability of NSSABC, some famous problems are applied in the simulation tests. Table 1 presents the brief descriptions of these benchmark functions.

In the experiment, the SN and the control parameter limit are set to 100. The threshold m is set to 5. For $D = 30$ and $D = 100$, the MAX FEs is equal to $1.5E+05$ and $5.0E+05$, respectively.

Table 2 gives the comparison results of four ABC algorithms when $D = 30$. From the results, NSSABC surpasses other three ABCs on all test cases except for f_3 . NSSABC finds the global optimal solution on the three test functions f_6 , f_9 , and f_{11} . For f_1 and f_2 , NSSABC greatly improves the quality of solutions comparing with IABC and GABC. However, GABC outperforms NSSABC on f_3 . NSSABC, GABC and IABC achieve the same result on both f_6 and f_9 . Overall, NSSABC provides better solutions than GABC, ABC and IABC.

Table 3 displays the results of four ABCs when $D = 100$. Like $D = 30$, these four algorithms do not achieve good results on f_3 , f_4 , and f_5 . When $D = 100$, ABC falls into local minima, but GABC, IABC and NSSABC can still find the global optimal solution. In general, for $D = 100$, the test results of these four ABC algorithms are worse than those of $D = 30$. For f_9 and f_{11} , NSSABC can still find the global optimal solution, but IABC and GABC fail to do.

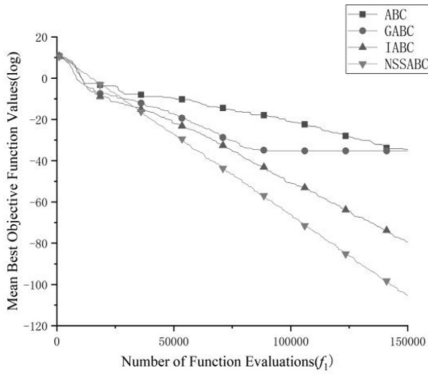
Table 2. Results achieved by ABC, IABC, GABC and NSSABC when $D = 30$

Fun	ABC		IABC		GABC		NSSABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	1.14E-15	3.58E-16	1.67E-35	6.29E-36	4.52E-16	2.79E-16	1.56E-46	4.67E-46
f_2	1.49E-10	2.34E-10	3.09E-19	3.84E-19	1.43E-15	3.56E-15	1.53E-25	3.28E-25
f_3	1.05E+04	3.37E+03	5.54E+03	2.71E+03	4.26E+03	2.17E+03	4.45E+03	6.59E+03
f_4	4.07E+01	1.72E+01	1.06E+01	4.26E+00	1.16E+01	6.32E+00	7.86E+00	1.28E+00
f_5	1.28E+00	1.05E+00	2.36E-01	3.94E-01	2.30E-01	3.72E-01	1.03E-01	2.42E-01
f_6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	1.54E-01	2.93E-01	4.23E-02	3.02E-02	5.63E-02	3.66E-02	1.67E-02	8.78E-03
f_8	-12490.5	5.87E+01	-12569.5	1.31E-10	-12569.5	3.25E-10	-12569.5	2.84E-10
f_9	7.11E-15	2.28E-15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{10}	1.60E-09	4.32E-09	3.61E-14	1.76E-14	3.97E-14	2.83E-14	2.19E-14	1.87E-14
f_{11}	1.04E-13	3.56E-13	0.00E+00	0.00E+00	1.12E-16	2.53E-16	0.00E+00	0.00E+00
f_{12}	5.46E-16	3.46E-16	3.02E-17	0.00E+00	4.03E-16	2.39E-16	3.01E-17	0.00E+00

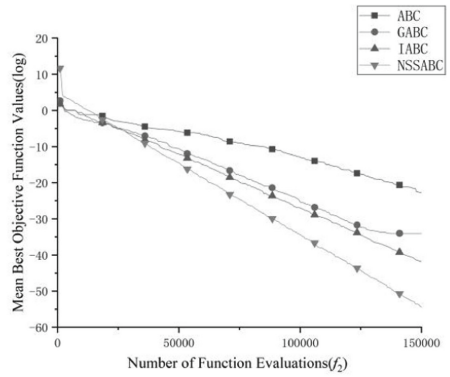
Table 3. Results achieved by ABC, IABC, GABC and NSSABC when $D = 100$

Fun	ABC		IABC		GABC		NSSABC	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	7.42E-15	5.89E-15	3.23E-33	1.45E-34	3.37E-15	7.52E-16	1.77E-44	8.64E-45
f_2	1.09E-09	4.56E-09	4.82E-18	3.53E-18	6.54E-15	2.86E-15	1.46E-23	1.19E-23
f_3	1.13E+05	2.62E+04	9.76E+04	2.81E+04	9.28E+04	2.71E+04	8.60E+04	2.23E+04
f_4	8.91E+01	4.37E+01	8.29E+01	1.28E+01	8.37E+01	3.68E+01	7.72E+01	2.58E+01
f_5	3.46E+00	4.29E+00	2.97E+00	2.72E+00	2.08E+01	3.46E+00	6.61E-01	5.64E-01
f_6	1.58E+00	1.68E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	1.96E+00	2.57E+00	7.45E-01	2.27E-01	9.70E-01	7.32E-01	3.93E-01	1.25E-01
f_8	-40947.5	7.34E+02	-41898.3	3.21E-10	-41898.3	5.68E-10	-41898.2	2.58E-10
f_9	1.83E-11	2.27E-11	1.42E-14	2.63E-14	1.95E-14	3.53E-14	0.00E+00	1.03E-14
f_{10}	3.54E-09	7.28E-10	1.50E-13	4.87E-13	1.78E-13	5.39E-13	1.35E-13	3.94E-13
f_{11}	1.12E-14	9.52E-15	7.78E-16	5.24E-16	1.44E-15	3.42E-15	0.00E+00	1.16E-15
f_{12}	4.96E-15	3.29E-15	9.05E-18	0.00E+00	2.99E-15	4.37E-15	9.04E-18	2.89E-15

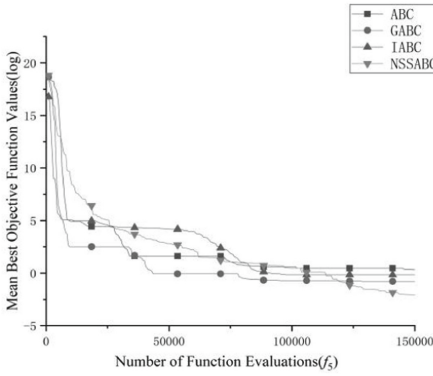
Figure 1 shows a comparison of the convergence performance of NSSABC and other ABCs. It is obvious that NSSABC is much faster than the other three algorithms except for f_5 , f_6 , and f_9 . The main reason is that the new search strategy based on GABC and IABC helps speed up the convergence on most of the test functions.



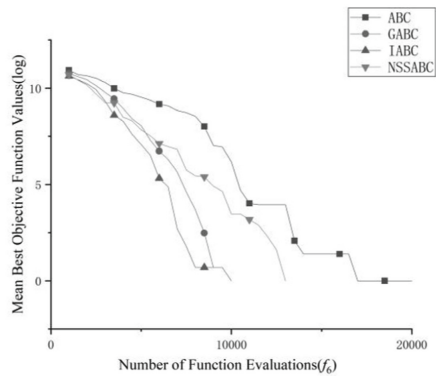
(a) Sphere (f_1)



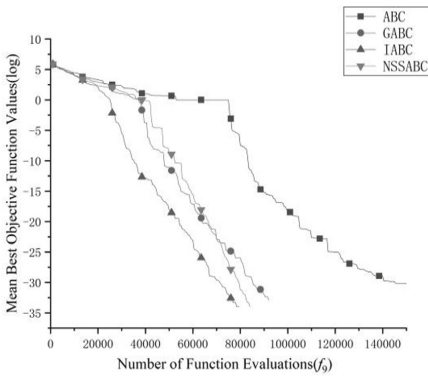
(b) Schwefel 2.22 (f_2)



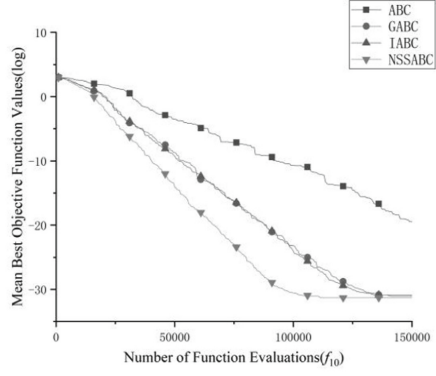
(c) Rosenbrock (f_5)



(d) Step (f_6)



(e) Rastrigin (f_9)



(f) Ackley (f_{10})

Fig. 1. The convergence characteristics of ABC, GABC, IABC and NSSABC on some functions for $D = 30$.

5 Conclusions

In this paper, we propose an improved ABC algorithm called NSSABC, which uses the information of the global best solution to guide the search. An external archive aims to preserve the updated global best solutions. It avoids the oscillating search to some extent. NSSABC can maintain good exploration and exploitation capacities. Experimental results show NSSABC algorithm is superior to the ABC, IABC, and GABC algorithms.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (No. 61663028).

References

1. Wang, M.: Research on micro grid optimal scheduling based on particle swarm optimization. *J. Ningxia Normal Univ.* **40**(10), 85–90 (2019)
2. Shi, J., Liu, Z., Pan, S.: Path planning improvement strategy of ant colony algorithm. *Fire Command Control* **44**(10), 153–162 (2019)
3. Zhang, M.X., Ma, X., Duan, Y.M.: Improved artificial bee colony. *J. Xidian Univ.* **42**(2), 65–70 (2015)
4. Du, Z.X., Liu, G.Z.: Artificial bee colony with global and unbiased search strategy. *Acta Electronica Sinica* **46**(2), 308–314 (2018)
5. Lin, Q., Zhu, M., Li, G.: A novel artificial bee colony with local and global information interaction. *Appl. Soft Comput. J.* **62**(3), 702–735 (2018)
6. Gao, W.F., Liu, S.Y., Huang, L.L.: A novel artificial bee colony algorithm with new probability model. *Soft. Comput.* **22**(7), 2217–2243 (2018)
7. Du, Z.X., Liu, G.Z., Zhao, X.H.: Integrated learning artificial bee colony algorithm. *J. Xi'an Univ. Electr. Sci. Technol.* **46**(2), 124–131 (2019)
8. Kucukkoc, I., Buyukozkan, K., Satoglu, S.I., Zhang, D.Z.: A mathematical model and artificial bee colony algorithm for the lexicographic bottleneck mixed-model assembly line balancing problem. *J. Intell. Manufact.* **30**(8), 2913–2925 (2019)
9. Zhu, G., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl. Math. Comput.* **217**(7), 3166–3173 (2010)
10. Wang, H., Wu, Z.J., Zhou, X.Y., Rahnamayan, S.: Accelerating artificial bee colony algorithm by using an external archive. In: *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 517–521 (2013)
11. Yurtkuran, A., Emel, E.: An adaptive artificial bee colony algorithm for global optimization. *Appl. Math. Comput.* **271**(10), 1004–1023 (2015)
12. Banharsakun, A., Sirinaovakul, B., Achalakul, T.: Job shop scheduling with the best -so-far ABC. *Eng. Appl. Artif. Intell.* **25**(2), 583–593 (2011)
13. Subotic, M., Tuba, M., Stanarevic, N.: Different approaches in parallel of the artificial bee colony algorithm. *Int. J. Math. Models Meth. Appl. Sci.* **5**(4), 755–762 (2011)

14. Tuba, M., Bacaninand, N., Stanarevic, N.: Guided artificial bee colony algorithm. In: Proceedings of the 5th European Conference on European Computing Conference (ECC 2011), pp. 398–403 (2011)
15. Li, G., Niu, P., Xiao, X.: Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. *Appl. Soft Comput. J.* **12**(1), 320–332 (2011)
16. Bi, X., Wang, Y.: An improved artificial bee colony algorithm. In: Proceedings of the 3rd International Conference on Computer Research and Development (ICCRD), pp. 174–177 (2012)
17. Gao, W.F., Liu, S.Y.: A modified artificial bee colony algorithm. *Comput. Oper. Res.* **39**(3), 687–697 (2012)
18. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **13**(5), 945–958 (2009)