Artificial Bee Colony Algorithm Based on Improved Search Strategy

Sumin Li[1], Weiyao Zhang[1], Jiatao Hao[1], Ruixiang Li[1](✉), and Juan Chen[2]

[1] School of Information Engineering, Minzu University of China, Beijing 10081, China
{smli,22302039,21011640}@muc.edu.cn, ruixiang0822@163.com
[2] China National Software and Service Company Limited, Beijing 10081, China

Abstract. The standard Artificial Bee Colony (ABC) algorithm exhibits slow convergence speed and a tendency to get trapped in local optima under certain circumstances. To overcome these limitations, researchers have proposed a new ABC algorithm (GABC) by using a modified search strategy. During the process of searching for solutions, the GABC algorithm incorporates some randomly selected individuals and the global best individual. However, the GABC algorithm still has drawbacks such as low search accuracy and slow convergence speed. In response to these issues, an improved artificial bee colony algorithm (IABC) is proposed in this paper. The IABC algorithm introduces a dynamic inertia weight factor based on the GABC algorithm. A set of standard test functions are used to test the optimization of the improved artificial bee colony algorithm. Experimental results demonstrate that the proposed algorithm outperforms both the standard ABC algorithm and the GABC algorithm in terms of search accuracy and convergence speed.

Keywords: Artificial bee colony · the global best individual · dynamic inertia weight factor

1 Introduction

The Artificial Bee Colony (ABC) algorithm was introduced by Karaboga in 2005 as a novel optimization algorithm based on the foraging behavior of honeybees. Its purpose was to solve multi-variable function optimization problems [1]. In 2014, Zhang proposed an improved version of the standard ABC algorithm to address its shortcomings such as slow convergence speed and premature convergence. The improved algorithm utilizes a random dynamic local search operator to perform local search on the current best food source, thereby accelerating the convergence speed. Additionally, a selection probability based on sorting is employed instead of relying solely on fitness to maintain population diversity and avoid premature convergence [2]. In 2015, Pan et al. proposed a new ABC algorithm (GABC) by using a modified search strategy inspired by a particle update model designed for Particle Swarm Optimization (PSO) by Mahmoodabadi et al. in 2014. Experimental results demonstrated that this improvement enhances the exploration capability of the ABC algorithm [3]. In 2018, Jin et al. drew inspiration from the Multi-Elite Artificial Bee Colony Optimization algorithm and introduced elite individuals and the global best individual in the bee colony to enhance the exploration of global optimal solutions [4]. In the same year, Chen et al. applied a tournament selection strategy to replace the original roulette wheel selection method to improve the issue of premature convergence. They also changed the replacement method for unchanged individuals, introducing a proportional replacement of individuals with the same optimal value. This approach preserves the current best solution while also having the ability to escape local optima [5]. In 2021, Su utilized a group collaboration mechanism in the ABC algorithm. During the employee bee phase, a large-step neighborhood search strategy was employed to enhance the exploration capability of the solution space. In the onlooker bee phase, a small-step neighborhood search strategy was used to improve the convergence accuracy of the solutions [6]. Wang et al. proposed an improved ABC algorithm in 2021 by enhancing the generation method of initial solutions and modifying the operations of three types of bees. These modifications aimed to enhance the optimization ability and robustness of the algorithm, as well as address the problem of getting trapped in local optima [7]. In 2022, Zhang et al. addressed the issue of poor stability in the ABC algorithm by introducing a chaotic initialization strategy. They applied a dynamic hybrid search strategy during the employee bee and onlooker bee phases to improve the algorithm's search traversal. Additionally, they incorporated elite solution information during the onlooker bee phase to balance global exploration and local exploitation capabilities [8]. Ren et al. combined the crossover mechanism with the global optimal-guided ABC algorithm to enhance the colony's exploration capability. They introduced the concept of sensitivity in the onlooker bee selection strategy to increase population diversity and avoid local optima [9]. Wang et al. replaced the onlooker bee search scheme with a probability-based reverse learning during the onlooker bee phase of the ABC algorithm. This modification effectively improved the optimization speed and convergence accuracy of the algorithm [10].

To overcome the issues of low search accuracy and slow convergence speed in the GABC, this paper proposes an Improved Artificial Bee Colony algorithm (IABC) based on the modified search strategy introduced in reference [3]. The IABC algorithm incorporates a dynamic inertia weight factor to enhance both global and local search capabilities. Furthermore, new search strategies are employed in the employed bee phase and onlooker bee phase to improve the algorithm's search performance. The proposed algorithm is compared with the standard ABC algorithm and the GABC algorithm using eight standard test functions, and its search performance is validated through experiments.

2 Artificial Bee Colony

The principle of the ABC is inspired by the foraging behavior of honeybees. During foraging, they first send out some bees to explore the surrounding environment and search for new food sources. After a bee finds a food source, it communicates the information to other bees, who then join in collecting the food. Throughout this process, bees continuously update their positions to achieve optimal foraging efficiency.

In the ABC algorithm, the search space is considered as a beehive, with numerous bees representing potential solutions. Each bee's position corresponds to the variable values of a solution. The algorithm consists of three types of bees: employed bees, onlooker bees, and scout bees.

Employed bees: Employed bees are responsible for searching for new solutions around their current positions. Each employed bee randomly selects a new position near its current position and calculates corresponding fitness value. If the fitness value of the new position is better than the current position, the bee updates its position to the new position.

Onlooker bees: Onlooker bees select employed bees to follow based on a roulette wheel selection rule. They then search for new solutions around the chosen employed bee's position and calculate their fitness value. The onlooker bees greedily choose the best solution and replace their own position if it is superior.

Scout bees: If a bee fails to find a better solution within a specified threshold, indicating it may be trapped in a local optimum, it becomes a scout bee. Scout bees randomly fly to new positions and calculate their fitness value to explore alternative solutions.

In the ABC algorithm, the bees' positions represent variable values of potential solutions, and their fitness values represent the quality of those solutions. The goal of the ABC algorithm is to continuously update the bees' positions to search for the global optimal solution.

The basic process of the ABC algorithm is as follows:

Initialization: Randomly generate a set of initial solutions known as the bee population. Each bee represents a solution x_{id}, which can be expressd as:

$$x_{id} = L_d + \text{rand}(0, 1)(U_d - L_d) \tag{1}$$

where x_{id} is the component of the i-th solution in the d-th dimension, $i \in \{1, 2, \ldots, SN\}$, SN is the number of bees, U_d and L_d represent the upper and lower bounds of the d-th dimension in the feasible solution space, respectively. rand$(0, 1)$ is a random number between 0 and 1.

Employed Bees Phase: Each employed bee explores the neighborhood of its current solution. They search for new solutions within the vicinity of their current solution using Eq. (2) based on the quality evaluation of the current solution.

$$x_{id}^{\text{new}} = x_{id} + \alpha \times \varphi\big(x_{id} - x_{jd}\big) \tag{2}$$

where d is the dimension of the current position, x_{id}^{new} is the new solution searched, $j \in \{1, 2, \ldots, SN\}$, $j \neq i$, which means a bee not equal to i is randomly selected among SN bees, φ is a random number uniformly distributed between -1 and 1, which determines the perturbation degree, and α is the acceleration coefficient (typically set to 1).

Onlooker Bees Phase: Bees exchange information among themselves. Then, based on Eq. (3), the onlooker bees calculate the selection probabilities to determine the probability of following each employed bee.

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

The onlooker bees choose employed bees to follow based on the selection probabilities p_i and generate new solutions in the vicinity of the employed bees using Eq. (2). The fitness values of the new solutions are evaluated, and the onlooker bees select the better solution based on a greedy strategy.

Scout Bees Phase: This phase determines if there are any solutions that need to be abandoned. If such solutions exist, a random replacement solution is generated using Eq. (1). This step enhances the global search capability of the algorithm.

Termination Criteria: Steps 2 to 4 are iteratively executed until a termination criterion is met, such as reaching the maximum number of iterations or finding a satisfactory solution.

The artificial bee colony algorithm can be used to solve various optimization problems in different fields, such as signal, image, and video processing, topology optimization, and artificial intelligence. Some of the real-world application scenarios for the artificial bee colony algorithm are:

Image segmentation: The artificial bee colony algorithm can be used to partition an image into meaningful regions based on some criteria, such as color, intensity, or texture.

Feature selection: The artificial bee colony algorithm can be used to select a subset of features from a large set of features that are relevant for a specific task, such as classification or clustering.

Path planning: The artificial bee colony algorithm can be used to find the optimal or near-optimal path for a robot or a vehicle to move from one location to another while avoiding obstacles and minimizing the cost.

3 Improved Artificial Bee Colony

The ABC algorithm is a swarm intelligence algorithm that has many advantages. However, one of its drawbacks is that each bee selects the optimal solution based on Eq. (2), which relies on the quality of the surrounding bees. This can result in slow convergence and susceptibility to local optima. To effectively tackle the aforementioned problems, Pan et al. introduced an enhanced ABC algorithm called GABC. This algorithm incorporates a new search strategy that includes a combination of randomly selected individuals and the global best individual to enhance the algorithm's exploration capability and accelerate convergence speed. The improved search strategy proposed by Pan is as follows:

$$x_{id}^{\text{new}} = \text{gbest}_d + \phi_{id} \cdot \left(2 \cdot \text{gbest}_d - x_{id} - x_{jd}\right) \tag{4}$$

where d is the dimension of the current position, x_{id}^{new} is the new solution searched, gbest_d is the global optimum. x_{id} is the component of the i solution in the d-th dimension, ϕ_{id} is a random number uniformly distributed between -1 and 1.

The difference between Eq. (4) and Eq. (2) lies in using the weighted average of three positions to update the velocity, instead of only considering the current position and a random position. This modification may lead to faster convergence of the algorithm by increasing the diversity in the search space. However, it may also result in unstable convergence performance since using three positions to update the velocity can cause the bees to jump too far in the search space, making it difficult to converge stably to the optimal solution. The algorithm should focus on the bees' global search capability in the early stages of the search and shift towards their local search capability in the later stages. Therefore, the parameters of the particle swarm algorithm should not remain fixed, and the inertia weight factor should vary with the number of iterations.

It can be observed that the mentioned equation does not incorporate any dynamic adjustment of the inertia weight factor. To further improve the development efficiency and search capability of the algorithm, and to prevent it from getting trapped in local optima, this paper proposes an improved Artificial Bee Colony algorithm (IABC). This algorithm introduces a dynamic inertia weight factor into the search strategy of GABC, where the value of the inertia weight factor can affect the global and local search capabilities of the algorithm. Shi proposed a linearly decreasing inertia weight particle swarm algorithm in reference [11], where the inertia weight gradually decreases as the number of iterations increases. The paper linearly decreases the inertia weight from 0.9 to 0.4, resulting in significant improvement in algorithm performance. This approach allows the particle swarm algorithm to have better global search capability in the early stages and better local search capability in the later stages. The inertia weight is adjusted according to the following equation.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{T} \times t \tag{5}$$

where ω_{max} 、 ω_{min} is the maximum and minimum values within a range, respectively, T is the maximum number of iterations, and t is the current number of iterations.

Inspired by the literature [3] and literature [11], a new search strategy is designed in this paper as follows.

Building upon the insights from references [3] and [11], this paper proposes a novel search strategy as follows.

During the employed bee phase, the search strategy is enhanced as follows:

$$x_{id}^{\text{new}} = \text{gbest}_d + \omega \cdot \phi_{id} \cdot \left(2 \cdot \text{gbest}_d - x_{id} - x_{jd}\right) \tag{6}$$

During the onlooker bee phase, the search strategy is improved as follows:

$$x_{id}^{\text{new}} = \omega \cdot \text{gbest}_d + \phi_{id} \cdot \left(2 \cdot \text{gbest}_d - x_{id} - x_{jd}\right) \tag{7}$$

In the initial stages of the algorithm, a larger value of ω is used, and it is dynamically adjusted linearly during the iteration process to gradually decrease its value. This ensures that the bees explore the solution space quickly with larger steps in the early iterations. In the later stages of the algorithm, a smaller value of ω is used to converge more precisely to the global optimum. This approach balances the convergence speed and accuracy while increasing the algorithm's robustness and flexibility.

The implementation steps for the improved algorithm are as follows:

Step 1: Initialize the population of solutions based on Eq. (1).
Step 2: Calculate the fitness value for each bee in the population.
Step 3: Repeat the following steps until the termination condition is met:
Step 4: Employed bees generate new solutions based on Eq. (6), calculate their fitness values, and select the solution based on a greedy strategy. If it is a better solution, replace the previous solution.
Step 5: Calculate the probabilities of employed bees being followed based on Eq. (3).
Step 6: Onlooker bees select employed bees to follow using the roulette wheel selection rule. They search for new solutions near the selected bee based on Eq. (7), calculate the fitness value of the new solution, and choose the solution based on a greedy strategy. If it is a better solution, replace the previous solution.

Step 7: If a bee fails to find a better solution within a certain threshold, it is abandoned. Scout bees then search for better solutions based on Eq. (1).
Step 8: Record the best solution until the final condition is met and output the best solution.

These steps outline the process of the improved algorithm, which incorporates a dynamic inertia weight factor and a modified search strategy to enhance its convergence speed and search capability, while addressing the limitations of the original Artificial Bee Colony algorithm.

4 Experiment

4.1 Test Functions

To validate the effectiveness of the ABC algorithm, the experiment used eight standard test functions defined in reference [12] as fitness functions for optimization testing. These eight functions encompass four basic characteristics: unimodal separable, unimodal non-separable, multimodal separable, and multimodal non-separable. The expressions, search ranges, and theoretical optimum values for each function are shown in Table 1. Among them, f_1, f_5, and f_6 have unimodal separable properties, f_2, f_3, and f_4 have unimodal non-separable properties, f_7 has multimodal separable properties, and f_8 has multimodal non-separable properties.

4.2 Test Functions Algorithm Accuracy Comparison

This study conducted simulation experiments using MATLAB 2016b environment. In this experiment, the proposed IABC was compared with the standard ABC algorithm and the GABC algorithm. The dimensions were set to 30 and 50, respectively. The population size of bees was set to 50, the maximum iteration count was 1000, and the threshold for determining the number of times trapped in local optima was set to 100. The results were obtained as the average of 10 experiments, and the average fitness values of the functions were recorded for 10 trials. The experimental result data can be found in Table 2 and Table 3.

According to the data in Table 2, for the 30-dimensional experiments, the IABC algorithm outperformed both the standard ABC algorithm and the GABC algorithm in all tested functions except for f_5. In function f_5, both the GABC and IABC algorithms achieved the same result by finding the global optimum. The standard ABC algorithm failed to find the global optimum in all tested functions, while the GABC algorithm only found the global optimum in function f_5. On the other hand, the IABC algorithm found the global optimum in functions f_1, f_2, and f_5.

As the dimension increases, the amount of data to be processed in each iteration also increases, and the information about the global optimum becomes more complex, making the optimization process more challenging. According to the data in Table 3, with increasing dimensions, the IABC algorithm still outperforms the standard ABC and GABC algorithms in all functions. Additionally, the convergence accuracy of the standard ABC and GABC algorithms decreases significantly, while the IABC algorithm

Table 1. Test function.

Functions	Domain	Global optimum
$f_1(x) = \sum_{i=1}^{D} x_i^2$	[−100, 100]	0
$f_2(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	[−100, 100]	0
$f_3(x) = \max_i(\lvert x_i \rvert 1 \le i \le D)$	[−100, 100]	0
$f_4(x) = \sum_{i=1}^{D-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$	[−30, 30]	0
$f_5(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$	[−100, 100]	0
$f_6(x) = \sum_{i=1}^{D} i x_i^4 + \text{rand}[0, 1)$	[−1.28, 1.28]	0
$f_7(x) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	[−5.12, 5.12]	0
$f_8(x) = -20 \cdot exp\left(-0.2 \cdot \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - exp\left(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + e$	[−32, 32]	0

Table 2. Test results when dimension 30.

Functions	ABC	GABC	IABC
f_1	3.86E + 03	1.95E-35	0
f_2	4.28E + 04	4.91E-02	0
f_3	4.57E-01	6.73E + 01	1.09E-215
f_4	1.01E + 07	6.84E + 01	2.79E + 01
f_5	4.90 E + 03	0	0
f_6	3.97E + 00	6.28E-03	1.56E-05
f_7	2.69E + 02	1.09E + 02	3.46E + 01
f_8	1.26E + 01	3.97E-10	4.44E-15

maintains a relatively high level of optimization accuracy. For functions f_1, f_2, and f_5, the IABC algorithm can still converge to the global optimum fitness. The experimental

Table 3. Test results when dimension 50.

Functions	ABC	GABC	IABC
f_1	3.11E + 04	1.80E-18	0
f_2	7.21E + 04	5.91E + 03	0
f_3	8.36E + 01	8.41E + 01	4.45E-86
f_4	1.36E + 08	9.13E + 01	4.76E + 01
f_5	3.41E + 04	1	0
f_6	8.46E + 01	1.36E-02	5.50E-05
f_7	5.37E + 02	2.43E + 02	1.98E + 02
f_8	1.87E + 01	1.37E + 00	4.44E-15

results demonstrate that the IABC algorithm has clear advantages over the standard ABC and GABC algorithms in high-dimensional optimization problems.

Based on the experimental data, the IABC algorithm shows generally favorable optimization performance for functions with the four basic properties. Moreover, regardless of low or high dimensions, the proposed algorithm achieves higher search accuracy compared to the other two algorithms.

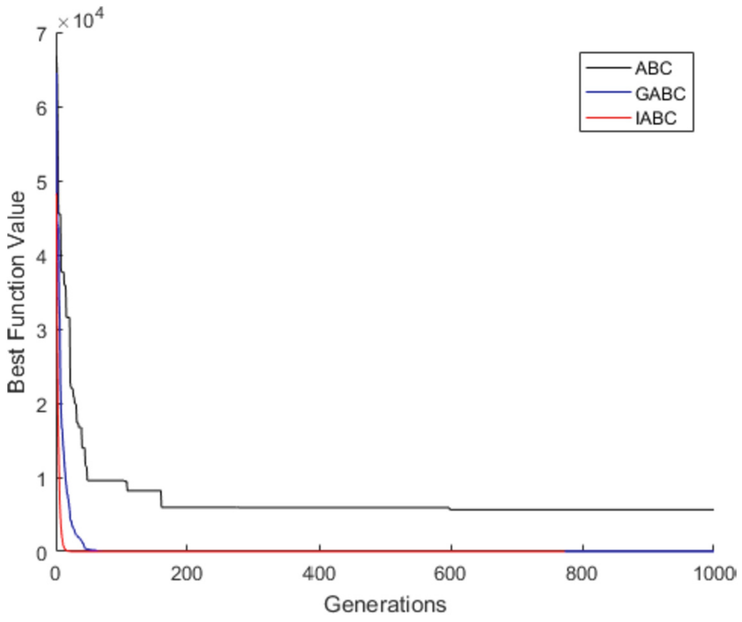

Fig. 1. The convergence of ABC, GABC and IABC on function f_1 when D = 30

4.3 Convergence Rate Comparison

To visually illustrate the search performance and convergence speed of the proposed algorithm, partial optimization curves are selected for several test functions. Figures 1, 2, 3, and 4 show the variations of the optimal function values with the number of search iterations for the functions f_1, f_2, f_7 and f_8 under the standard ABC, GABC, and IABC algorithms. These four test functions represent different basic properties, where f_1 represents unimodal separable properties, f_2 represents unimodal non-separable properties, f_7 represents multimodal separable properties and f_8 represents multimodal non-separable properties. Figures 1 and 2 depict the convergence curves for a dimension of 30, while Figs. 3 and 4 represent the convergence curves for a dimension of 50.

f_1

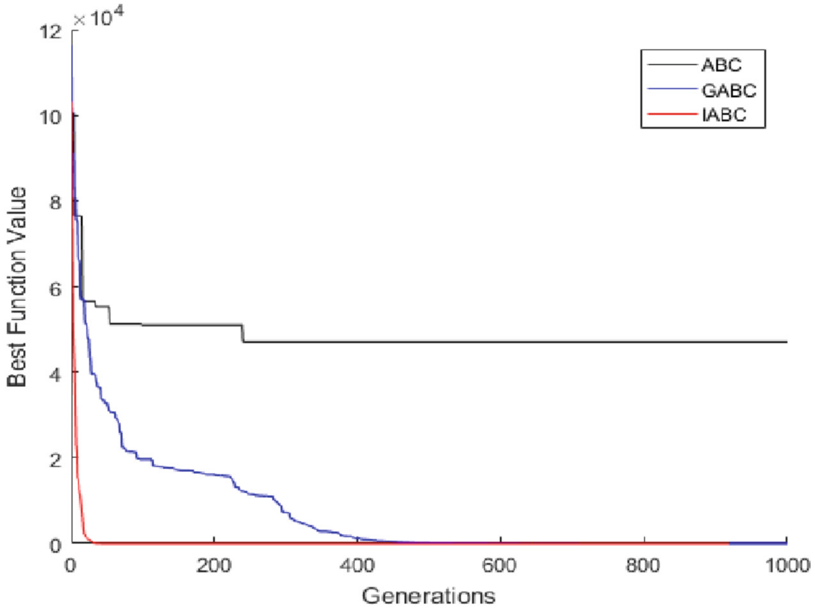

Fig. 2. The convergence of ABC, GABC and IABC on function f_2 when $D = 30$

From Fig. 1, it can be observed that when dealing with a unimodal separable function f_1, the proposed algorithm does not exhibit a significant advantage in terms of speed compared to the GABC algorithm, which also demonstrates fast convergence and accuracy. However, Fig. 2 reveals a significant improvement in convergence speed for the proposed algorithm when dealing with the unimodal non-separable function f_2.

Figures 3 and 4 demonstrate that when facing complex multimodal functions, the proposed algorithm shows significant improvements in both optimization accuracy and convergence speed. It achieves fast optimization while ensuring high accuracy. Overall, the optimal function value curves of the IABC algorithm are steeper compared to the standard ABC and GABC algorithms, indicating better search results and a noticeable improvement. The IABC algorithm consistently discovers better solutions at a faster rate.

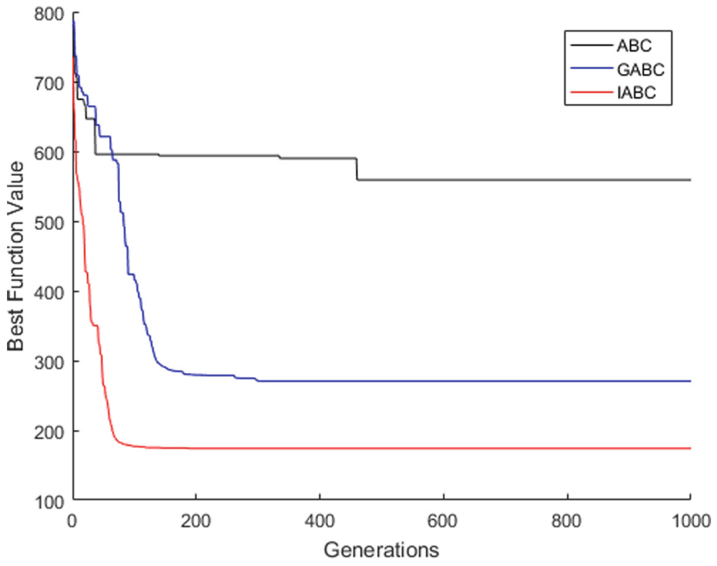

Fig. 3. The convergence of ABC, GABC and IABC on function f_7 when $D = 50$

In summary, the IABC algorithm exhibits higher optimization efficiency and better exploratory capabilities. It demonstrates good convergence performance and the ability to find more accurate solutions for both unimodal and multimodal test functions.

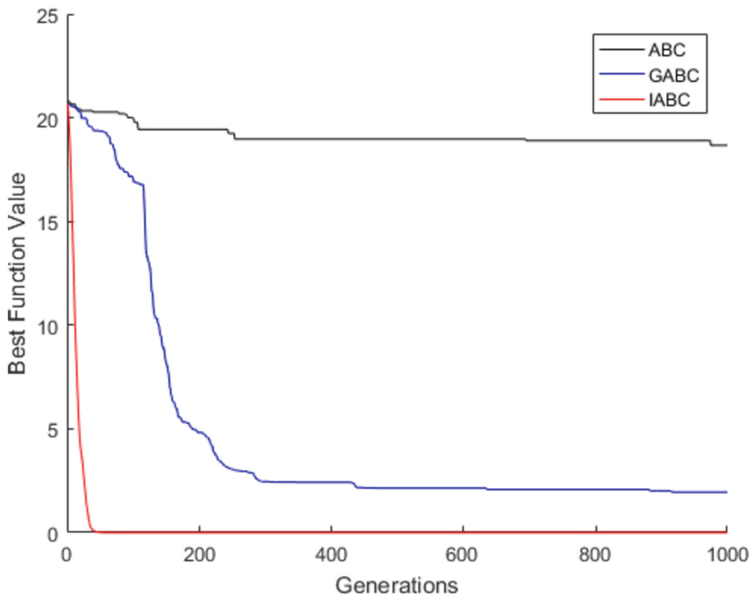

Fig. 4. The convergence of ABC, GABC and IABC on function f_8 when $D = 50$

5 Conclusion

This paper proposes an improved ABC algorithm (IABC) that introduces a dynamic inertia weight factor based on GABC. Additionally, new search strategies are employed in both the employed bee phase and the onlooker bee phase to replace the original search strategy. Experimental results demonstrate that the introduction of the dynamic inertia weight factor effectively enhances the algorithm's global search capability, preventing it from getting trapped in local optima. The improved ABC algorithm significantly accelerates the convergence speed and achieves convergence values that are closer to the global optimal values of the test functions.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grant 62176273 and by National first-class undergraduate major in software engineering.

References

1. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Erciyes University (2005)
2. Zhang, P.: Research on Bayesian Network Structure Learning Based on Artificial Bee Colony Algorithm. Xi'an University of Electronic Science and Technology (2014)
3. Pan, X.Q., Lu, Y., Li, S.M., Li, R.X.: An improved artificial bee colony with new search strategy. Int. J. Wirel. Mob. Comput. **9**(4), 391–396 (2015)

4. Jin, Y., Sun, Y., Wang, J., Wang, D.: Improved elite artificial bee colony algorithm based on simplex method. J. Zhengzhou Univ. (Eng. Sci. Edn.) **39**(06), 36–42 (2018)
5. Chen, S., Ji, W., Qiu, Y., Zhang, G.: Improved artificial bee colony algorithm for solving flexible job-shop scheduling problem. J. Mach. Tools Autom. **05**, 161–164 (2018)
6. Su, M.: Improved Artificial Bee Colony Algorithm and Its Application Research. Zhongyuan Institute of Technology (2021)
7. Wang, Y., Ma, M., Ge, J., Miao, S.: Flexible job shop scheduling based on improved artificial bee colony algorithm. J. Mach. Tools Autom. **03**,159–163+168 (2021)
8. Zhang, H., Long, D., Qin, T., Wang, X., Yang, J.: Improved artificial bee colony algorithm for WSN coverage and connectivity optimization. Comput. Eng. Des. **43**(10), 2701–2710 (2022)
9. Ren, J., Du, Z., Wang, X.: Improved artificial bee colony algorithm for cloud task scheduling. J. Henan Univ. Sci. Technol. (Nat. Sci. Edn.) **43**(04), 55–60+6–7 (2022)
10. Wang, J., Wang, B., Ge, M.: Artificial bee colony algorithm based on reverse learning. J. Mudanjiang Normal Univ. (Nat. Sci. Edn.) **01**, 23–30 (2022)
11. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Compution.Washington, pp. 1945–1950. IEEE (1999)
12. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. **3**(2), 0–102 (1999)