IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 25, 2020, accepted January 20, 2021, date of publication January 25, 2021, date of current version February 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054179

Improved Bat Algorithm for UAV Path Planning in Three-Dimensional Space

XIANJIN ZHOU[1], FEI GAO[1], XI FANG[1], AND ZEHONG LAN[2]
[1]School of Science, Wuhan University of Technology, Wuhan 430070, China
[2]School of Economics and Management, Wuhan University, Wuhan 430072, China

Corresponding author: Fei Gao (gaof@whut.edu.cn)

This work was supported in part by the Equipment Pre-Research Ministry of Education Joint Fund under Grant 6141A02033703 and in part by the Fundamental Research Funds for the Central Universities under Grant 2018IB017.

ABSTRACT This paper describes the flight path planning for unmanned aerial vehicles (UAVs) based on the advanced swarm optimization algorithm of the bat algorithm (BA) in a static environment. The main purpose of this work is that the UAVs can obtain an accident-free, shorter, and safer flight path between the starting point and the endpoint in the complex three-dimensional battlefield environment. Based on the characteristics of the standard BA and the artificial bee colony algorithm (ABC), a new modification of the BA algorithm is proposed in this work, namely, the improved bat algorithm integrated into the ABC algorithm (IBA). The IBA mainly uses ABC to modify the BA and solves the problem of poor local search ability of the BA. This article demonstrates the convergence of the IBA and performs simulations in MATLAB environment to verify its effectiveness. The simulations showed that the time required for the IBA to obtain the optimum solution is approximately 50% lower than the BA, and that the quality of the optimum solution is about 14% higher than the ABC. Furthermore, by comparing with other traditional and improved swarm intelligent path planning algorithms, the IBA can plan a faster, shorter, safer, accident-free flight path for UAVs. Finally, this article proves that IBA also has good performance in optimizing functions and has broad application potential.

INDEX TERMS Battlefield environment, path planning, improved bat algorithm, convergence, local search.

I. INTRODUCTION

As life and military needs continue to grow, unmanned aerial vehicles (UAVs) play an increasingly important role in many areas. Compared with manned aircraft, UAVs have the advantages of high optical resolution, short warning time, low cost, and high maneuverability [1]. Therefore, UAVs typically perform dangerous, boring, complex tasks in a variety of areas [2], [3]. In the history of UAVs development, path planning has been thought of as a key factor in the process of performing tasks. A path with strong security, good feasibility, high computational efficiency, and low cost can greatly improve the efficiency of completing the UAVs missions [4].

In fact, planning the flight path of UAVs usually requires optimization algorithms to optimize the flight path. Optimization methods generally fall in deterministic mathematical programming methods and stochastic metaheuristic algorithms [5]. However, deterministic methods of mathematical programming are prone to stagnation in non-linear space research, which requires high preparation for mathematics [6]. Over the past years, stochastic metaheuristic algorithms have been increasingly used to solve UAV path planning problems due to their flexibility, simplicity, and ability to avoid local optimization. In general, stochastic metaheuristic algorithms may be classified into evolutionary, physics-based, and swarm intelligence (SI) [7]. Evolutionary algorithms generally generate better new populations through combinations and mutations between earlier generations of individuals, such as Genetic Algorithm (GA) [8], Differential Evolution Algorithm (DE) [9], and Biogeography-based Optimization (BBO) [10], *et al.* Physics-based methods are to use rules extracted from different physical phenomena in nature in search of objectives [6]. Some well-known algorithms are the Simulated Annealing Algorithm (SA) [11], Gravity Cable Algorithm (GSA) [12], Central Force Optimization Algorithm (CFO) [13]. SI algorithms usually mimic and foraging activities of animals in nature. It can save the solution obtained so far, uses fewer operators, and is easy to implement than the evolutionary algorithm [7]. Therefore, SI is more widely used in UAV path planning problems. Popular SI includes Artificial Bee Colony Algorithm (ABC) [14], Particle Swarm Optimization Algorithm (PSO) [15],

The associate editor coordinating the review of this manuscript and approving it for publication was Xiwang Dong.

Ant Colony Algorithm (ACO) [16], [18], Bat Algorithm (BA) [19], [20], *et al*.

However, stochastic metaheuristic algorithms also have unavoidable disadvantages. David H.Wolpert and William G.Macready [21] proposed no free lunch (NFL) theorems in 1997. They logically proved that no metaheuristic algorithm could best solve all optimization problems. In other words, an intelligent algorithm can obtain the desired result in a particular optimization problem, but it misbehaves in other problems. Therefore, people are trying to integrate different intelligent algorithms into UAV path planning to find better solutions. In terms of the theoretical design of the controller, in [22], a 6-degree of freedom nonlinear PID controller (NLPID), which combines the GA, is designed to meet the system stability and tracking requirements of a four-wing UAV. The improved active disturbance rejection control (IADRC) proposed in [23], [24] can stabilize and suppress external interference and system uncertainty, and minimize the control energy, adjustment time, and steady-state error. The decentralized control scheme based on IADRC also provides good performance [25]. In addition, other new controllers have been designed recently, and can better solve the problems in the corresponding fields, such as a consistent control system for three quadrotor intelligent bodies [26], a new classic adaptive controller based on a synergetic theory [27], a model-free active input-output feedback linearization technique based on IADRC [28]. On the other hand, a large number of papers show that improved swarm intelligence algorithms can also better solve the flight path planning problem. Cristian Ramirez Atencia proposes a new weighted random generator that reduces the convergence speed of the multi-objective evolutionary algorithm (MOEA) [9]. The algorithm based on disturbed fluid and trajectory proposed by Yao Peng can effectively avoid obstacles to a certain extent [4]. Moreover, there is a large volume of published studies showing that improved intelligent algorithms can succeed in route planning. For instance, an aging-based ant colony optimization algorithm (ABACO) is proposed in [29], which considers the aging of the individual and releases different pheromones according to different ages. [30] conducts further research on [29] and solves the path planning problem of a dynamic environment. Despite there are many types of intelligent algorithms in the planning of UAVs, they suffer from different main drawbacks. For example, in the process of planning the flight path, the lack of mutation mechanism of standard BA is easy to fall into local optimum, resulting in the population losing subsequent evolutionary capacity.

In order to solve the problem of poor local search ability of BA, which was first proposed by Xin-She Yang in 2010 [31], there has been an increasing amount of literature on improved BA in recent years. People proposed a new directional bat algorithm, which improved the classic bat algorithm in four ways and greatly improved the performance of BA in [32]. Amir H. Gandomi and Xin-She Yang tried to combine the BA algorithm with chaos, which uses four different variations to replace the invariant parameters in the BA, and is verified by thirteen different chaotic maps [33]. Trong-The Nguyen proposed the bat-bee colony algorithm (BA-ABC) [34]. The algorithm mainly iterates the results of the BA algorithm and the ABC algorithm, then replaces the better results of the both parties with the poor results of the other to realize the evolution of the group. In [35], a hybrid particle swarm optimization-improved frequency bat algorithm (PSO-MFB) and obstacle detection and avoidance algorithm (ODA) are proposed, which effectively solves the path planning problem in a dynamic environment. After that, [36] further improved the algorithm of [35] and proposed a conflict-free shortest path planning algorithm. Furthermore, in the field of path planning, in order to increase the diversity of the population, Gaige Wang applied the bat algorithm with a mutation to UAVs path planning [18]. N.Lin's studies have reported that the enhanced artificial potential field method combined with the chaotic bat algorithm may enhance the robustness of the algorithm [37].

Although the standard BA can provide a better quality solution, it takes a lot of time. However, the ABC can quickly obtain the solution, and the quality of the solution is poor. The main contribution of this paper is to propose a new algorithm to solve the problem of UAVs flight path planning, which mainly combines the characteristics of BA and ABC to achieve the purpose of improving the local search ability and obtaining a crash-free, safer and shorter flight path. The IBA algorithm proposed in this article mainly contains two modules. The first module involves the generation of points, which is implemented through the BA. In order to improve the local search ability, the mutation factor is taken into consideration. Then, ABC is used to modify the results of the first module, so as to further enhance the local search ability of the algorithm.

The overall structure of the study takes the form of six chapters, including this introductory chapter. Section 2 introduces the mathematical model of 3D space in UVA path planning. In section 3, the principle of the classic BA is described. Subsequently, BA with mutation added ABC for UAV path planning is presented and its convergence is proven in detail in section 4. The fifth chapter tests 9 parameters that appear in the IBA algorithm, compares IBA with other swarm intelligence algorithms, and uses IBA to solve UAVs path planning and benchmark function optimization problems. The final section gives a summary and identifies areas for further research.

II. UAV MATHEMATICAL MODEL

In the history of the development of UAVs, path planning has been thought of as a key factor in the process of performing tasks. This chapter mainly describes the comprehensive cost model of UAVs under different threats and the selection method of random path nodes.

A. PATH PREPROCESSING

To accelerate the convergence of the algorithm, the initial UAVs path planning is shown in Fig.1. It is assumed that UAV

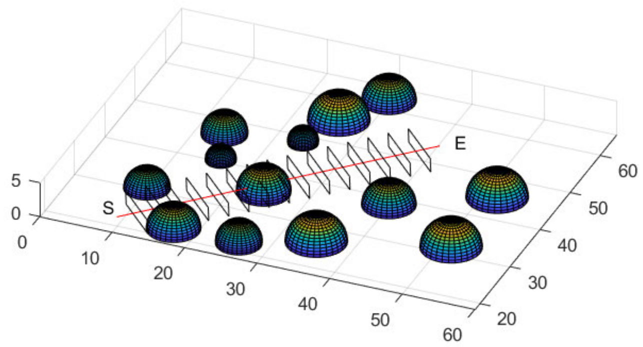


FIGURE 1. 3D battlefield environment model.

needs to fly from the starting point $S(x_0, y_0, z_0)$ to the ending point $E(x_E, y_E, z_E)$. There are some threat areas during the flight. We convert the original start and end points to new coordinates in the x-axis direction by using Eq.1, where (x, y, z) represents the original coordinates, (x', y', z') is the rotated coordinates, θ is the angles between line SE and XOY plane, and φ is the angle between the projection of SE on the XOY plane and the X-axis.

$$tan\theta = \frac{|z_E - Z_0|}{\sqrt{(x_E - x_0)^2 + (y_E - y_0)^2}}$$
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} cos\varphi cos\theta & -sin\varphi & -cos\varphi sin\theta \\ sin\varphi cos\theta & cos\varphi & -sin\varphi sin\theta \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \tag{1}$$

Then, we divide the rotated line segment SE into $(D + 1)$ segments (include S and E) and passing each node except S and E to make planes $L_1, L_2, \ldots, L_D$ perpendicular to the straight line SE [18], [38], [39]. After that, randomly select a point on each plane. Obviously, we can get D points and connect them with S node and E node. So that the three-dimensional path planning problem can be turned into a D-dimensional function optimization problem.

B. COMPREHENSIVE COST MODEL

The battlefield environment is complex and changeable with various threats such as radar, climate, missiles, anti-aircraft guns, *et al*. These threats will affect the mission completion of UAV's missions. In addition, UAV maneuverability is also a non-negligible poppy that affects path selection.

The scope of the threat on the battlefield is typically dependent on the combination of different cylindrical or conical geometry [1]. Assuming that the ranges of various threats are spherical areas with different radius. Eq.2 is the probability of radar threat detecting UAVs [39].

$$P(d_R) = \begin{cases} 0, & d_R > d_{R_{max}} \\ \frac{1}{d_R^4}, & d_{R_{min}} \le d_R \le d_{R_{max}} \\ 1, & d_R < d_{R_{min}}. \end{cases} \tag{2}$$

where $d_{R_{min}}, d_{R_{max}}$ are the minimum and maximum range of the radar threat. d_R is the distance between UAV and radar source.

In order to facilitate the experimental simulation, the climatic threats, anti-aircraft threats, missile threats and the probability of destroying the UAV are as follows:

$$P_i(d_i) = \begin{cases} 0, & d_i > d_{i_{max}} \\ \frac{1}{d_i}, & d_{i_{min}} \le d_i \le d_{i_{max}} \\ 1, & d_R < d_{i_{min}}. \end{cases} \tag{3}$$

where $P_{A'}(d_{A'}), P_M(d_M), P_C(d_C)$ are the threat probability of antiaircraft guns, missiles, and atmosphere to UAV. d_i is the distance from the UAV to the threat source. $d_{i_{min}}, d_{i_{max}}$ are the minimum and maximum range of the threat.

Apart from these threats, UAV also threatens to crash when flying over mountains. Supposing that the terrain model is composed of several mountains with different center positions, and the mountains are approximately replaced by cones. Eq.4 is the expression of mountain height [1].

$$z_i(x, y) = h_i e^{-\frac{(x-a_i)^2}{10} - \frac{(y-b_i)^2}{10}} \tag{4}$$

h_i is the height of the mountain and (a_i, b_i) is the center of the mountain. If the flying height is lower than the mountain height, the probability of UAV $(P_T(d_T))$ being destroyed is 1. Conversely, $P_T(d_T)$ is 0.

According to the maneuverability of the UAVs, this paper considers the constraints of fuel consumption and maximum climb angle. Fuel consumption is usually measured by flight distance. Assume there are n segments in the path and each segment is l_i. l_{max} is the maximum path length. Therefore, the path constraints are:

$$\sum_{i=1}^{n} l_i \le l_{max} \tag{5}$$

This article primarily simulates the path planning of UAVs in 3D space. In order to better integrate the real situation, we consider that the maneuvering performance can affect the maximum angle of climb and altitude limitation. Therefore, this paper assumes the maximum climb angle is 45° and the maximum height of the flight is 6 kilometers. When the UAV's path is beyond the maximum climb angle, the probability of the UAV crash $P_B = 1$. On the contrary, $P_B = 0$. Similarly, When the UAV's flight altitude exceeds the maximum flight altitude limit, the probability of the

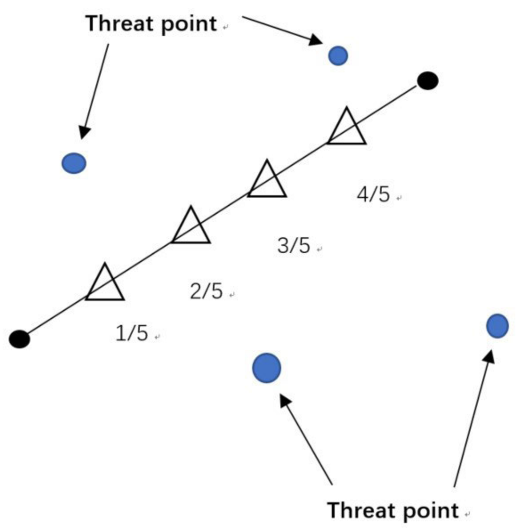


FIGURE 2. Sub-path calculation method.

UAV crash $P_H = 1$. On the contrary, $P_B = 0$. The comprehensive track cost can be measured by using Eq.6.

$$minW = min \int_0^L \sum \delta w(s)ds \tag{6}$$

where $\sum \delta w(s) = \delta_O w_O(s) + \delta_R w_R(s) + \delta_M w_M(s) + \delta_C w_C(s) + \delta_{A'} w_{A'}(s) + w_B(s) + w_H(s)$. L is track path length. W is optimization objective function. $w_O(s)$, $w_R(s)$, $w_M(s)$, $w_{A'}(s)$, $w_C(s)$, $w_T(s)$, $w_H(s)$ are the cost of path, radar threat, missile threat, anti-aircraft threat, climate threat, terrain threat, maximum climb angle and maximum height. $\delta_O, \delta_R, \delta_M, \delta_{A'}, \delta_C$ are the weight of each threat cost and their sum is 1.

For simplicity, each path is divided into five segments on average and the threat cost is calculated at the end of each discrete segment (as shown in Fig.2). At last, the average value of the discrete segments is assumed to be the threat cost of this segment. We can calculate the cost according to the following Eq.7.

$$\begin{aligned} w_{L_{i,j}} &= \frac{1}{5}\sum\nolimits_{k=1}^{5} w_{k,L_{i,j}} \\ w_{k,L_{i,j}} &= \delta_R P_R(d_R) + \delta_M P_M(d_M) + \delta_{A'} P_{A'}(d_{A'}) \\ &\quad + \delta_C P_C(d_C) + P_T(d_T) + P_B(d_B) + P_H(d_H) \end{aligned} \tag{7}$$

where $L_{i,j}$ represents the process of UAV flying from node i to node j. $w_{k,L_{i,j}}$ is the threat cost of UAV at the $k-th$ point of the sub-segment.

III. CLASSICAL BAT ALGORITHM

The classic bat algorithm is a swarm intelligence algorithm. Its search strategy is inspired by the social behavior of bats and the use of echo in foraging and avoiding obstacles. Besides, the bat algorithm is a promising algorithm, which combines the advantages of PSO, GA, and harmony search algorithm to a certain extent [31].

In nature, some bats not only use echolocation, but also use their vision and smell to find food and avoid obstacles. Even the loudness and frequency emitted by bats are constantly changing. For simplicity, we idealize some of the echo characteristics of the bat and follow the rules [18].

(1) All bats only use echolocation to perceive distance, then find targets and avoid obstacles.

(2) Bats can automatically adjust the wavelength and frequency of their transmitted pulses while preying on prey. They fly randomly at position X_i with speed V_i, fixed frequency f_{min}, loudness A_0, and continuously adjust the pulse transmission frequency $r \in [0, 1]$ depending on the proximity to the target.

(3) The bat loudness varies from the smallest constant value A_{min} to A_0.

This study set out to simulate three-dimensional space UAV flight. Thus, we use Eq.8 to define the update rule for the ith bat's speed V_i^t, frequency f_i and new solution X_i^t at time step t.

$$\begin{aligned} f_i &= f_{min} + (f_{max} - f_{min})\beta \\ V_i^t &= V_i^{t-1} + (X_i^{t-1} - X^*)f_i \\ X_i^t &= X_i^{t-1} + V_i^t \end{aligned} \tag{8}$$

where $\beta \in [0, 1]$ is a random number. Here, X^* is the optimal solution from time step 1 to time step $t-1$, and then X^* is only updated when all bats have determined their position in time step t. Generally speaking, the frequency of the ultrasonic waves emitted by each bat is different. Therefore, each bat was randomly assigned a frequency $f_i \in [0, 100]$.

For the local search section, we generate a random number $rand_1 \in [0, 1]$. Once $rand_1 > r_i$, the new solution X_{new} to replace the original solution X_i^t, which is obtained by randomly walking on the current optimal solution.

$$X_{new} = X^* + \varepsilon A^t \tag{9}$$

where random number $\varepsilon \in [-1, 1]$, while A^t is the average loudness of all bats at time step t.

Furthermore, we create another random number $rand_2 \in [0, 1]$. If $rand_2 < A_i^t$ and the fitness of the new solution $f(X_i^t)$ is less than the fitness of the current optimal solution $f(X^*)$, A_i^{t+1} and r_i^{t+1} are updated on the basis of Eq.10.

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t \\ r_i^{t+1} &= r^0[1 - exp(-\gamma t)] \end{aligned} \tag{10}$$

where α, γ, r^0 are constants. Based on the above analysis, the main part of the classic bat algorithm is described in Algorithm 1 and in Fig.3.

IV. IMPROVED BAT ALGORITHM (IBA)

This paper intends to integrate the advantages of ABC and mutation operators into the BA. ABC inspired by bee colony foraging behavior, used by Dervis Karaboga earlier and compared with other algorithms [14]. ABC is mainly divided into three steps [40], [41]. Beginning, bees randomly search for honey sources. Bees with high quality honey sources

Algorithm 1 BA Algorithm

Begin:

1: Initialization. Set the number of bat populations N_P, the maximum number of iterations T_{max}, initialize the loudness A^0, initial speed V^0, frequency r, constant α, γ, and generate counter $t = 1$ of each bat.

2: Calculate the fitness of each bat $f(X_i^0)$

3: For $t = 1 : T_{max}$

4: For $i = 1 : N_P$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X^*)f_i$$
$$X_i^t = X_i^{t-1} + V_i^i$$

5: If $rand_1 > r_i^t$, generate a new solution X_{new} instead of X_i^t

$$X_{new} = X^* + \varepsilon A^t$$

End if

6:Calculate the fitness of the new solution $f(X_i^t)$.

7: If $rand_2 < A_i^t$ and $f(X_i^t) < f(X^*)$

Accept the new solutions and update r_i^t, A_i^t.

End if

End for

8: Update the current optimal solution X^*.

End for

9: Choose the optimal solution as the final result.

End

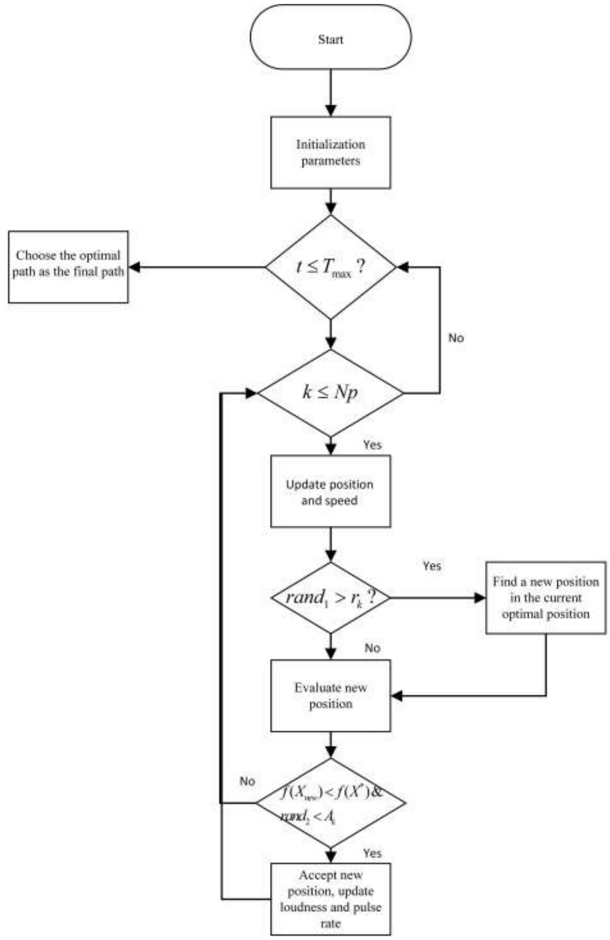


FIGURE 3. Flow chart of bat algorithm.

are called employed bees and bees with poor quality honey sources are called onlooker bees. Then, the employed bees recruit the onlooker bees at the honey source and search together near the honey source. If an improved honey source is found, the original honey source is replaced. Otherwise, the honey source remains unchanged. Finally, if the sub-optimal honey source does not improve over a period of time, the employed bees become a scout bee and randomly searches for the honey source to replace the original honey source. Repeat these three steps continuously until the maximum number of repetitions is reached.

Generally speaking, we convert the three-dimensional problem of UAVs track planning into a D-dimensional function optimization problem, which means that each bat can represent a planning path. The main idea of IBA is that the bat population updates position X_i by using speed V_i, loudness A_i, frequency f_i, *et al.* and then the local position is changed by using the characteristics of ABC.

To better combine the advantages of the BA and the ABC, we have improved the behavior of individuals. Each individual generates a random solution. The first part of the individual with a smaller fitness is selected as the employed bee and the remainder as the onlooker bee. Information update method of employed bees is based on the standard BA's steps, namely through V_i^t, f_i, A_i^t, *et al* to update the solution X_i^t. Then, the onlooker bees choose to employed bees through roulette and Eq.11.

$$p(i) = \frac{f(X_i)}{\sum_{k=1}^{N_e} f(X_j)} \tag{11}$$

where $f(X_i) = 1/W(X_i)$ is the fitness function of the ith individual and N_e is the number of employed bees.

For onlooker bees behavior, we introduce mutation factor F [18] to enhance the local search ability of the algorithm and only change a certain node of the individual each time (except for the start and endpoints).

$$X_{i,j} = X_{r_1,j} + F(X_{r_2,j} - X_{r_3,j}) \tag{12}$$

where $X_{i,j}$ is $j-th$ vector of the ith onlooker bee. Random number r_1, r_2, r_3 are employed bee serial number and $X_{r_1} \neq X_{r_2} \neq X_{r_3} \neq X_i$, In order to improve the iterative speed of the algorithm, we only calculate the path cost of the path before and after the replacement point.

To enhance the local search ability of the algorithm, when third random number $rand_3 > r_i^t$, local search based on Eq.13.

$$X_{i,j} = X_j^* + \varepsilon A^t \tag{13}$$

where X_j^* is the $j-th$ vector of the current optimal solution. Through the greedy criterion, the best result is chosen to replace the original path.

The detailed IBA algorithm steps are as follows:

(1). Initialize the loudness A_i, frequency $f_i \in [f_{min}, f_{max}]$, and speed V_i of each bat in the population, and generate a random solution $X_i^0 = [S, x_{i_1}, x_{i_2}, \ldots, E] \in [X_{min}, X_{max}]$.

(2). Calculate the fitness of each bat $f(X_i^0)$, $1 \leq i \leq N_p$. All bats are arranged in order of fitness, select the bat with the smallest fitness as the current optimal X^*, the top 50% small bats are selected as employed bees, and the remaining bats are selected as onlooker bees.

(3). Update the speed V_i^t and position X_i^t of the employed bee according to Eq.8

(4). Generate a random number $rand_1$. If $rand_1 > r_i^t$, generate a new solution X_{new} instead of X_i^t through Eq.9

(5). Calculate the fitness of the solution $f(X_i^t)$.

(6). Generate a random number $rand_2$. If $rand_2 < A_i^t$ and $f(X_i^t) < f(X^*)$, accept the solution X_i^t, and update A_i^t and r_i^t by Eq.10.

(7). Repeat steps 3-6 to update all employed bees information.

(8). onlooker bees choose employed bees through roulette, and randomly select the node j that needs to be changed (except for the start and end points).

(9). Randomly select three solutions $r_1 \neq r_2 \neq r_3$ that are different from the onlooker bee, and update $X_{i,j}^t$ of the onlooker bees according to Eq.12

(10). Generate a random number $rand_3$. If $rand_3 > r_i^t$ and generate a new solution $X_{i,j}^t(new)$ near the optimal solution X^* to replace $X_{i,j}^t$ by Eq.13.

(11). Calculate the fitness of the onlooker bee. If the fitness of the onlooker bee is better than the corresponding employed bee, then the solution of the onlooker bees can replace the solution of the employed bee, otherwise, the solution of the employed bee cannot change.

(12). Repeat steps 8-11 until all onlooker bees information are updated.

(13). Update the optimal solution X^* again.

(14). If the solution of the employed bee X_i^t does not change after a certain period of time T_{limit} and is not the optimal solution X^*, a random solution $X_{new} \in [X_{min}, X_{max}]$ will be generated to replace the employed bee X_i^t, and initialize the corresponding speed V_i^t,frequency f_i, loudness A_i^t and pulse rate r_i^t.

(15). After the iteration is complete, choose the optimal solution as the final result. The main part of IBA is described in Algorithm 2 and Fig.4.

Although this article mainly uses the IBA algorithm to solve the UAV path planning problem in three-dimensional space, the theory proves that the convergence of the IBA algorithm is still very necessary. Analyzing the convergence of the algorithm theoretically can promote the improvement and development of the algorithm, and provide clear theoretical significance for the improvement of the algorithm. Similar to the proof in [44], we can get Theorem 1 Theorem 2, and Theorem 3. The definition 1-6 are stated in detail in Appendix.

Algorithm 2 IBA Algorithm

Begin:

1: Initialization. Set the number of bat populations N_P, the maximum number of iterations T_{max}, initialize the loudness A^0, initial speed V^0, pulse frequency r, constant α, γ, and generate counter $t = 1$ of each bat.

2: Calculate the fitness of each bat $f(X_i^0)$, choose a part as the employed bees N_e and the rest as the onlooker bees N_s.

3: For $t = 1 : T_{max}$

4: For $i = 1 : N_e$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X^*)f_i$$
$$X_i^t = X_i^{t-1} + V_i^i$$

5: If $rand_1 > r_i^t$, generate a new solution X_{new} instead of X_i^t

$$X_{new} = X^* + \varepsilon A^t$$

End if

6: Calculate the fitness of the new solution $f(X_i^t)$.

7: If $rand_2 < A_i^t$ and $f(X_i^t) < f(X^*)$

Accept the new solutions and update r_i^t, A_i^t.

End if

End for

8: Update the current optimal solution X^*.

9: For $i = 1 : N_s$

10: Onlooker bee selects employed bee through roulette and randomly determines the nodes j that need to be changed.

11: Randomly choose three different paths from the employed bee $r_1 \neq r_2 \neq r_3$ and Update the position of the employed bee node j

$$X_{i,j} = X_{r_1,j} + F(X_{r_2,j} - X_{r_3,j})$$

12: If $rand_3 > r_i^t$

$$X_{i,j} = X_j^* + \varepsilon A_t$$

End if

13: Calculate the path cost and the best path replaces the original path through the greedy criterion.

End for

14: Update the optimal solution.

15: Determine if any honey source is exhausted. If so, re-plan the path to replace the original path, and initialize the corresponding speed V_i^t, frequency f_i, loudness A_i^t and pulse rate r_i^t.

End for

16: Choose the optimal solution as the final result.

End

Theorem 1: In the IBA algorithm, the state sequence $\{S(t); t \geq 0\}$ of the group is a finite homogeneous Markov chain.

Proof: (1) In practical problems, the search space of any optimization algorithm is finite. In addition, the speed and spatial location of any individual can be

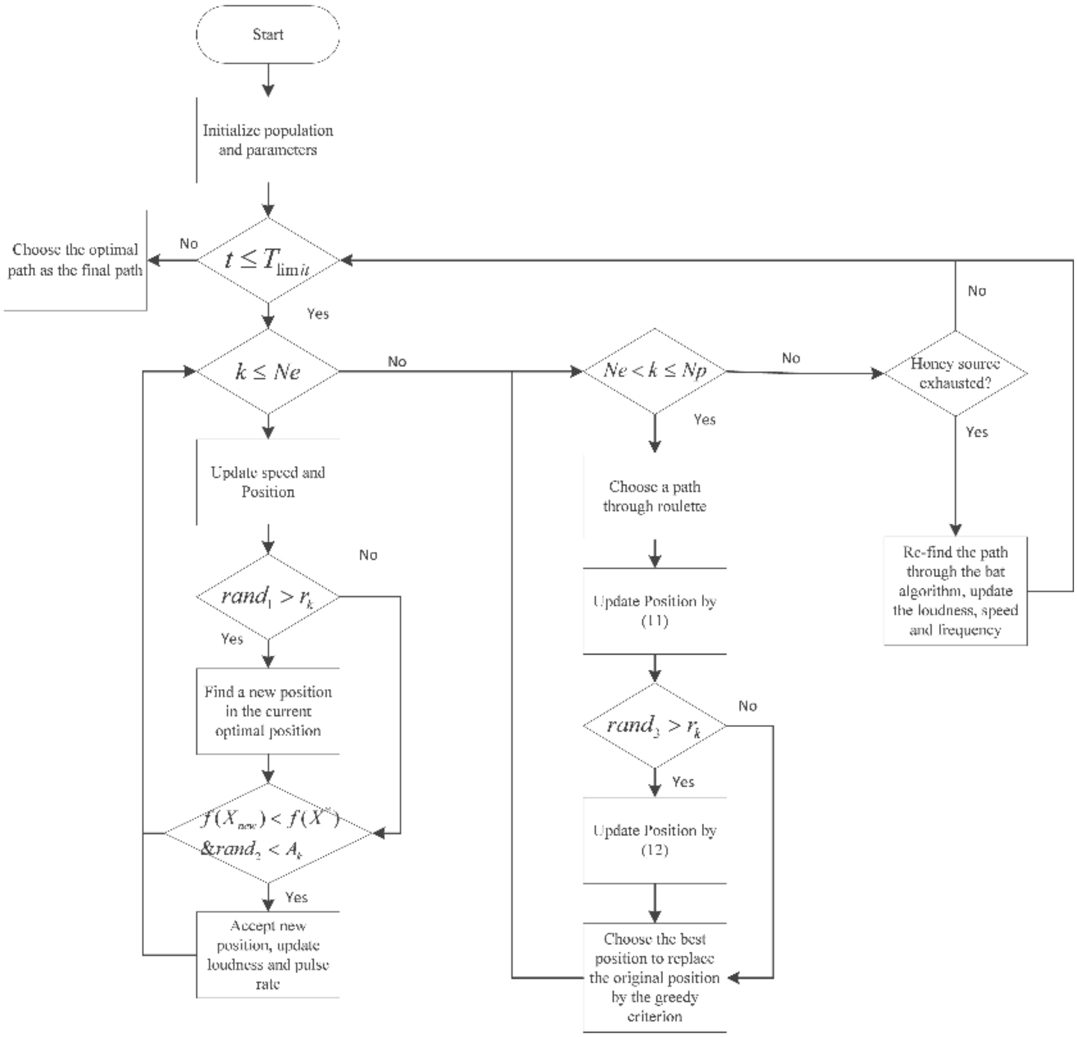


FIGURE 4. Improved bat algorithm flow chart.

TABLE 1. Parameter settings.

Name	Symbol	Value
Radar threat range	$d_{R_{min}}$	4km
	$d_{R_{max}}$	120km
Missile threat range	$d_{M_{min}}$	3.5km
	$d_{M_{max}}$	100km
Anti-aircraft threat range	$d_{A'_{min}}$	3km
	$d_{A'_{max}}$	6km
Climate threat range	$d_{C_{min}}$	2km
	$d_{C_{max}}$	7km
Path weight	δ_O	0.1
Radar threat weight	δ_R	0.3
Climate threat weight	δ_C	0.2
Missile threat weight	δ_M	0.2
Anti-aircraft threat weight	$\delta_{A'}$	0.2
Maximum turning angle	θ_{max}	45^O

limited, and the whole group is composed of N_p individuals. Therefore, the state space of the algorithm is finite.

TABLE 2. Compared with other swarm intelligence algorithms.

Algorithm	Minimum path cost	Mean path cost	Iteration time
IBA	1.2106	1.2133	1.5053
GFACO [1]	1.5809	1.8004	6.5917
ABC	1.3400	1.4259	1.8083
BA-ABC[34]	1.4031	1.4395	1.6336
BAM[18]	1.2687	1.3273	6.0919
IABC[41]	1.4072	1.4969	1.9860
BA	1.2341	1.2760	3.0487

(2) According to Definition 4 in Appendix, in the group state sequence $s(t) : t > 0$, for $\forall s(t-1), s(t) \in S$, $p(T_S(S(t-1)) = X(t))$ affects $p(T_S(s(t-1)) = s(t))$. Then from individual state transition probability we know that $p(T_S(s(t-1)) = X(t))$ is only related to the state which is at $t-1$, and is not related to time $t-1$. So, the state sequence $\{s(t) : t > 0\}$ is a finite homogeneous Markov chain.

TABLE 3. Statistical results of different populations, ratios of employed bees and onlooker bees.

Population	Employed bees: Onlooker bees	Minimum cost	Mean cost	Cost variance	Mean number of iterations	Mean time
10	2:3	1.2975	1.3242	0.0021	98.9400	0.3721
	1:1	1.3094	1.3447	0.0048	98.3400	0.3998
	3:2	1.3120	1.3281	4.9971e-04	98.1200	0.4320
	4:1	1.3304	1.3444	0.0023	96.7000	0.4992
20	1:4	1.2236	1.2278	7.7464e-05	99.2800	0.6032
	2:3	1.2765	1.2783	9.3971e-05	99.0400	0.7392
	1:1	1.2750	1.2938	8.5924e-05	98.7800	0.7990
	3:2	1.2695	1.2838	2.8268e-04	97.6600	0.8844
	4:1	1.2969	1.3093	3.6907e-04	95.6600	1.0126
30	1:4	1.2185	1.2250	4.8707e-05	99.5200	0.9076
	2:3	1.2574	1.2683	6.1119e-05	99.1200	1.1134
	1:1	1.2687	1.2728	1.1143e-05	97.9000	1.1907
	3:2	1.2756	1.2805	1.7039e-04	96.9200	1.2938
	4:1	1.2587	1.2637	2.3069e-04	95.0600	1.5235
40	1:4	1.2209	1.2230	3.5585e-05	99.3000	1.2193
	2:3	1.2532	1.2568	4.4730e-05	98.7000	1.4722
	1:1	1.2589	1.2642	8.2952e-05	97.9000	1.6097
	3:2	1.2531	1.2627	5.5883e-05	96.8000	1.7703
	4:1	1.2536	1.2648	4.0443e-04	94.7800	2.0689
50	1:4	1.2106	1.2133	4.3977e-05	99.2000	1.5053
	2:3	1.2434	1.2454	2.8465e-05	98.9200	1.8652
	1:1	1.2384	1.2469	1.8484e-04	97.9000	2.0445
	3:2	1.2499	1.2554	4.8768e-05	97.3400	2.2993
	4:1	1.2602	1.2729	1.8556e-04	94.2400	2.9673
70	1:4	1.1975	1.2050	6.3162e-05	99.2200	2.1447
	2:3	1.2353	1.2432	9.1851e-05	98.2600	2.7520
	1:1	1.2375	1.2447	1.0784e-06	97.7000	2.9192
	3:2	1.2403	1.2437	3.0574e-05	96.8200	3.2711
	4:1	1.2453	1.2490	8.2616e-05	92.4600	3.6524

TABLE 4. Statistical results of IBA and ABC for different T_{limit}.

T_{limit}	Statistics	Cost			Iteration time		
		IBA	ABC	IABC	IBA	ABC	IABC
10	Minimum	1.2084	1.3400	1.5141	1.5500	1.0470	1.1710
	Maximum	1.2307	1.6234	2.1957	1.8470	2.2910	1.5690
	Mean	1.2144	1.4259	1.7393	1.6211	1.8083	1.2824
	Variance	4.7960e-05	0.0033	0.0299	0.0020	0.0314	0.0025
20	Minimum	1.2133	1.3441	1.5052	1.4550	0.9980	1.1960
	Maximum	1.2339	1.5699	1.9700	1.7830	1.3570	1.6850
	Mean	1.2178	1.4347	1.6321	1.5285	1.0865	1.2829
	Variance	1.8427e-05	0.0027	0.0103	0.0023	0.0020	0.0040
40	Minimum	1.1980	1.3388	1.5084	1.4350	1.7350	1.1530
	Maximum	1.2169	1.5291	1.8611	1.7480	2.2490	1.6260
	Mean	1.2007	1.4324	1.6128	1.5245	1.8208	1.2699
	Variance	4.3757e-05	0.0027	0.0079	0.0024	0.0048	0.0036
80	Minimum	1.2076	1.3352	1.5303	1.4600	1.0490	1.1420
	Maximum	1.2270	1.5514	1.7747	1.9270	2.1680	1.6240
	Mean	1.2095	1.4156	1.6327	1.5229	1.1998	1.2454
	Variance	2.7826e-05	0.0029	0.0056	0.0040	0.0893	0.0037

Theorem 2: The optimal state set G composed of the optimal state of the individual is a closed set on the group state space S.

Proof: For $\forall S_i \in G, \forall S_j \notin G$, any step $l \geq 1$, we can get Eq.29 from Ckapman-Kolmogorov equation:

$$P^l_{s_i,s_j} = \sum_{s_{r_1}\in S} \cdots \sum_{s_{r_{l-1}}\in S} p(T_S(s_i) = s_{r_1}) \cdot p(T_S(s_{r_1}) = s_{r_2}) \cdots p(T_S(s_{r_{l-1}}) = s_j) \tag{14}$$

where $P^l_{s_i,s_j}$ is the probability of group state s_i transitioning to state s_j after l step. There is $p(T_S(s_{r_{a-1}}) = s_{r_a})$ in each product expression of the expansion of Eq.23, satisfying $s_{r_{a-1}} \in G, s_{r_a} \notin G$, where $1 \leq a \leq l$. By Definition 4,

$$p(T_S(s_{r_{a-1}}) = s_{r_a}) = \prod_{m=1}^{N_p} p(T_S(X_{i_m}) = X_{j_m}) \tag{15}$$

From $s_{r_{a-1}} \in G$ and $s_{r_a} \notin G$, there is $f(X_a) > f(X_{a-1}) = f(g^*) = inf(f(c)), c \in A$. Thus there is $p(T_S(s_{r_{a-1}}) = s_{r_a}) = 0$ at least, at this time $P^l_{s_i,s_j} = 0$. So, G is a closed set on S.

Theorem 3: The Markov chain population sequence of the IBA algorithm can converge to the global optimum with probability 1.

Proof: We can find from the introduction of the IBA that the evolution direction of the entire population is

TABLE 5. Statistical results of IBA, BA and BA-ABC for different r^0 and γ.

γ	r^0	Statistics	Cost			Iteration time		
			IBA	BA	BA-ABC	IBA	BA	BA-ABC
0.2	0.3	Mean	1.2075	1.2621	1.4107	1.6244	3.3984	1.5007
		Variance	1.5803e-05	4.1789e-04	0.0034	0.0034	1.2612	0.0081
	0.6	Mean	1.2131	1.2626	1.4313	1.6394	2.6224	1.4808
		Variance	4.6255e-05	9.0404e-04	0.0026	0.1395	0.1123	0.0121
	0.9	Mean	1.2141	1.2732	1.3775	1.5915	2.3846	1.4022
		Variance	6.4994e-05	5.7141e-04	2.91e-04	0.0026	0.1691	0.0402
0.5	0.3	Mean	1.2095	1.2595	1.4079	1.5346	2.5188	1.4649
		Variance	3.5200e-05	2.4257e-04	0.0045	0.0108	0.3146	0.0116
	0.6	Mean	1.2035	1.2603	1.4333	1.5773	3.2965	1.4273
		Variance	5.5020e-05	3.7594e-04	0.0019	0.0014	0.8916	0.0057
	0.9	Mean	1.2145	1.2836	1.4401	1.4509	2.2684	1.3012
		Variance	1.0424e-04	9.4497e-04	0.0052	0.0025	0.1371	0.0552
0.8	0.3	Mean	1.2040	1.2625	1.5436	1.5295	2.6118	1.5628
		Variance	1.0794e-04	5.7877e-04	0.4730	0.0013	0.2302	0.1419
	0.6	Mean	1.2157	1.2612	1.4186	1.5028	2.6020	1.3983
		Variance	3.3091e-05	4.0658e-04	0.0021	0.0014	0.1620	0.0119
	0.9	Mean	1.2239	1.2842	1.4202	1.7547	2.3614	1.2834
		Variance	1.5262e-05	4.8645e-04	0.0100	0.2448	0.1421	0.0551

monotonous. Assuming that the state $s(t)$ in which the group is in time t has entered the global optimal solution set G, then it is in state $s(t+1)$ at time $t+1$, and $P\{s(t+1) \in G| s(t) \in G\} = 1$ always holds. Thereby,

$$
\begin{aligned}
&P\{s(t+1) \in G\} \\
&= P\{s(t) \notin G\}P\{s(t+1) \in G|s(t) \notin G\} \\
&\quad + P\{s(t+1) \in G\}P\{s(t+1) \in G|s(t) \notin G\} \\
&= (1 - P\{s(t) \in G\})P\{s(t+1) \in G|s(t) \notin G\} \\
&\quad + P\{s(t) \in G\}
\end{aligned}
$$

Let $P\{s(t+1) \in G|s(t) \notin G\} \geq h(t) \geq 0$, $\lim_{t\to\infty} \prod_{i=1}^{t}(1 - h(i)) = 0$, then:

$$
\begin{aligned}
&1 - P\{s(t+1) \in G\} \\
&= 1 - (1 - P\{s(t) \in G\})P\{s(t+1) \in G|s(t) \notin G\} \\
&\quad - P\{s(t) \in G\} \\
&= (1 - P\{s(t) \in G\})(1 - P\{s(t+1) \in G|s(t) \notin G\}) \\
&\leq (1 - P\{s(t) \in G\})(1 - h(t)) \\
&\leq \prod_{i=1}^{t}(1 - h(i))(1 - P\{s(0) \in G\})
\end{aligned}
$$

So,

$$P\{s(t+1) \in G\} \geq 1 - \prod_{i=1}^{t}(1 - h(i))(1 - P\{s(0) \in G\})$$

When $t \to \infty$, there is:

$$\lim_{t\to\infty} P\{s(t+1) \in G\} \geq 1$$

However, $0 \leq P\{s(t+1) \in G\} \leq 1$. So,

$$\lim_{t\to\infty} P\{s(t+1) \in G\} = 1$$

Obviously, after iteration, the IBA can finally converge to the global optimal solution.

TABLE 6. Comparison of different intelligent algorithms with different F.

F	Statistics	IBA	DE	BAM	ABC
0.1	Mean path cost	1.2163	1.6802	1.3210	1.4208
	Minimum path cost	1.2130	1.4239	1.2648	1.3337
	Mean iteration time	1.5196	6.9255	5.9858	1.1060
0.2	Mean path cost	1.2148	1.4693	1.3092	1.4200
	Minimum path cost	1.2083	1.3804	1.2586	1.3323
	Mean iteration time	1.5260	6.7247	5.9813	1.1099
0.3	Mean path cost	1.2158	1.4057	1.3233	1.4217
	Minimum path cost	1.2036	1.3389	1.2480	1.3404
	Mean iteration time	1.5317	6.7502	5.9223	1.1025
0.4	Mean path cost	1.2093	1.3797	1.3269	1.4217
	Minimum path cost	1.2079	1.3247	1.2633	1.3386
	Mean iteration time	1.5377	6.7232	5.9093	1.1096
0.5	Mean path cost	1.2096	1.3864	1.3068	1.4243
	Minimum path cost	1.2094	1.2292	1.2612	1.3364
	Mean iteration time	1.4944	7.1077	5.8654	1.1175
0.6	Mean path cost	1.2173	1.4370	1.3037	1.4117
	Minimum path cost	1.2150	1.2579	1.2492	1.3393
	Mean iteration time	1.5257	6.7446	5.8716	1.1072
0.7	Mean path cost	1.2128	1.5115	1.3094	1.4149
	Minimum path cost	1.2047	1.3156	1.2437	1.3406
	Mean iteration time	1.5275	6.7947	5.9018	1.1039
0.8	Mean path cost	1.2134	1.5549	1.3060	1.4181
	Minimum path cost	1.2043	1.3537	1.2584	1.3390
	Mean iteration time	1.5023	6.5376	5.8568	1.1149
0.9	Mean path cost	1.2169	1.6610	1.311	1 1.4270
	Minimum path cost	1.2152	1.4715	1.2633	1.3441
	Mean iteration time	1.5159	6.2901	5.8583	1.0990

V. RESULT

A. UAV PATH PLANNING PROBLEM

In order to verify the effectiveness of the algorithm in UAVs path planning. This chapter not only compares and analyzes the various parameters of the IBA, but also compares the IBA algorithm with other intelligent algorithms and the extended algorithm of the BA. In addition, this paper conducts experimental simulation based on MATLAB R2019b software with the computer processor Intel Core i5 2.40GHz, RAM 16.00GB, and 64-bit Windows 10 operating system.

In order to be closer to the real environment, this work constructs a three-dimensional flight environment. We sets

TABLE 7. Statistical results of iba and ba for different A^0 and α.

A^0	α	Statistics	Cost		Iteration time	
			IBA	BA	IBA	BA
0.95	0.9	Mean	1.2126	1.2760	1.4795	3.0487
		Variance	3.8588e-05	8.1993e-04	0.0025	0.9132
	0.7	Mean	1.2162	1.2649	1.4795	2.9449
		Variance	5.3072e-05	8.3617e-04	0.0026	0.5441
	0.5	Mean	1.2217	1.2613	1.4787	2.4256
		Variance	2.3148e-05	3.9987e-04	0.0021	0.2611
	0.3	Mean	1.2319	1.2669	1.4629	2.6481
		Variance	8.1194e-05	7.1842e-04	0.0015	0.1206
0.55	0.9	Mean	1.2268	1.2381	1.5618	2.4034
		Variance	2.5066e-05	1.5984e-04	0.0021	0.2002
	0.7	Mean	1.2318	1.2371	1.5924	2.1888
		Variance	1.9721e-05	9.0073e-05	0.0019	0.3000
	0.5	Mean	1.2472	1.2428	1.4808	2.2532
		Variance	5.3312e-05	3.4018e-04	0.0022	0.3116
	0.3	Mean	1.2553	1.2392	1.4831	2.3531
		Variance	6.5747e-05	1.7847e-04	0.0017	0.3330
0.15	0.9	Mean	1.2728	1.2783	1.4913	2.3938
		Variance	2.1907e-05	0.0229	0.0027	0.1577
	0.7	Mean	1.2759	1.3034	1.4988	2.3745
		Variance	1.0842e-05	0.0246	0.0016	0.2296
	0.5	Mean	1.2829	1.2600	1.5621	2.4423
		Variance	2.2550e-04	0.0147	0.0014	0.1305
	0.3	Mean	1.2959	1.3018	1.6229	2.4582
		Variance	2.3596e-04	0.0334	0.0047	0.1618

TABLE 8. Comparison of different intelligent algorithms with different *D*.

D	Statistics	IBA	BA	ABC	PSO	BAM
10	Mean path cost	0.9580	0.9916	1.1519	1.4271	1.3353
	Minimum path cost	0.9598	1.0213	1.0807	1.1284	1.0638
	Mean iteration time	1.3174	1.5988	1.1018	3.2113	4.9385
15	Mean path cost	1.2759	1.3674	1.4710	2.2013	2.4720
	Minimum path cost	1.2588	1.3136	1.4001	1.4735	1.6410
	Mean iteration time	1.7285	2.5621	1.8445	3.5792	6.9151
20	Mean path cost	1.5933	1.7785	1.8577	3.0489	2.6242
	Minimum path cost	1.5893	1.6014	1.7512	1.8952	4.5399
	Mean iteration time	1.8062	4.8021	1.8369	5.6430	8.3847
25	Mean path cost	1.9442	2.6224	2.2603	4.1518	6.2990
	Minimum path cost	1.9330	2.2651	2.0784	2.3613	3.5261
	Mean iteration time	2.0760	4.6426	1.1953	5.9015	10.4059
30	Mean path cost	2.2966	2.6307	2.7173	5.4352	8.6961
	Minimum path cost	2.2850	2.7119	2.4361	3.4344	6.5801
	Mean iteration time	3.3284	6.0821	1.9276	7.1851	13.3736

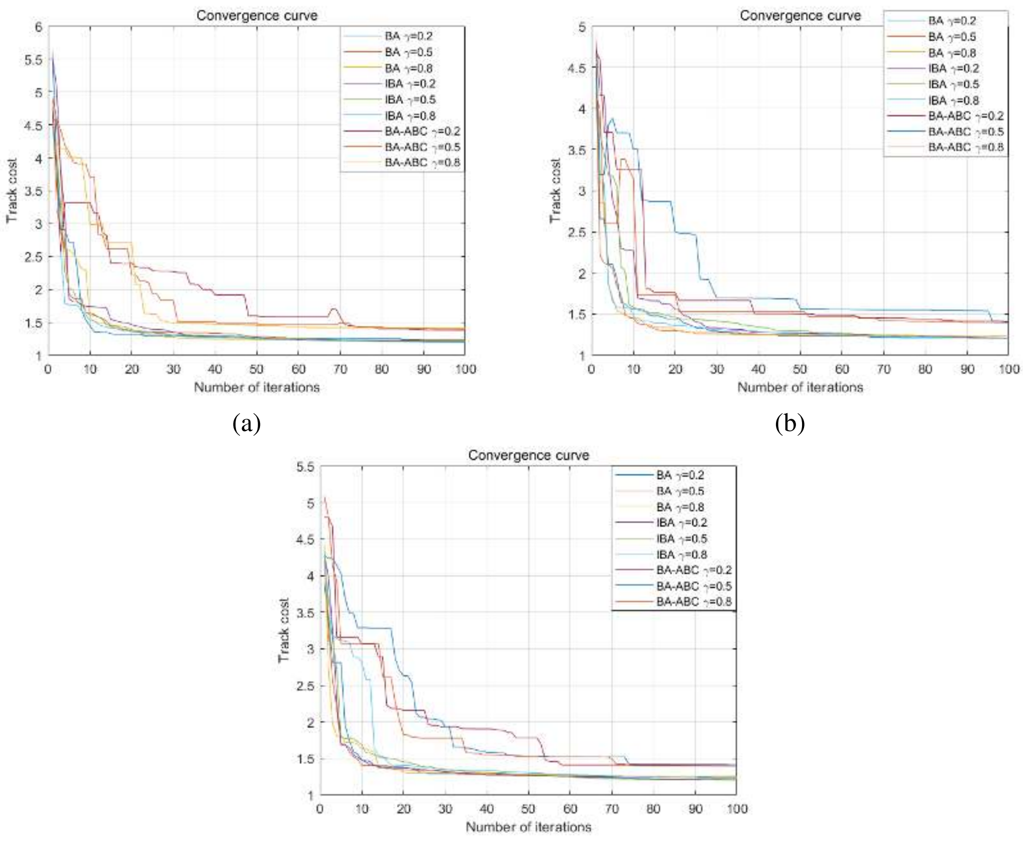
the starting node $S(10, 20, 0)$, the target node $E(42, 50, 2.8)$, the radar threat point $R_1(26, 55, 0.2)$, $R_2(35, 26, 0.2)$, $R_3(35, 26, 0.2)$, $R_4(51.5, 31, 0.4)$, the missile threat point $M_1(17, 22, 0.2), M_2(24, 35, 0.4), M_3(30, 62, 0.2)$, the artillery threat point $A'_1(17, 22, 0.2)$, $A'_2(22, 26, 0.4)$, $A'_3(14, 46, 0.6)$, the climate threat point $C_1(16, 40, 0.4)$, $C_2(24, 48, 0.6)$. In addition, we set the parameters in Table 1 [1]:

From Table 2 we can find that compared with other swarm intelligence algorithms, the IBA greatly reduces the iteration time and can get a better optimal solution.

It is very necessary to explore the influence of parameters on the performance of IBA. The IBA proposed in this paper deals with establishing multiple parameters. So we have therefore conducted many experiments to determine the appropriate range of parameters. At the same time, other parameters remain consistent to ensure fairness in the experience. Furthermore, when the optimum value remains unchanged, the algorithm is thought to have achieved the convergence value. In the experiment, when the parameters are set, we begin to record the iteration time of the algorithm. This article conducts 50 simulations experiments for each comparison test. Table 3 shows the influence of different population numbers and different ratios of employed bees and onlooker bees of the IBA.

From Table 3 we can find when the population increases or the proportion of employed bees is large, the iteration time of the IBA can also increase. The main reason is that the search rules of employed bees and onlooker bees are different, and employed bees take longer to search for paths. In this article, the IBA is more suitable for the population size range of 20-50, and the ratio of employed bees to onlooker bees is 1:4. Within this range, IBA may obtain better results more quickly.

This article then tests parameters T_{limit}, F, r^0, and γ in the IBA algorithm in turn, and compares the IBA algorithm with other intelligent algorithms horizontally and vertically.



(a) (b) (c)

FIGURE 5. The optimal solution convergence curve of BA algorithm and IBA algorithm with different γ and r^0 (a) $r^0 = 0.3$; (b) $r^0 = 0.6$; (c) $r^0 = 0.9$.

The statistics from the simulation experiments in Table 4-6 show that the parameters T_{limit}, F, r^0, and γ have a little impact on the performance of IBA, especially in the average path cost and convergence time that we are concerned about.

Then, we test the influence of parameters A^0 and α on the IBA. Table 7 shows that A^0 and α have no obvious influence on the iteration time of IBA, but the larger the value of A^0 and α, the smaller the average path cost obtained. In other words, the flight path for the UAVs under the IBA is better.

At last, take $T_{limit} = 10$, $A^0 = 0.95$, $\alpha = 0.9$, $F = 0.5$, $r^0 = 0.6$, $\gamma = 0.5$ as an example, we explore the impact of the final parameter, namely, the number of nodes D, on the new algorithm.

From the experimental results in Table 8, the average iteration time and average path cost of the algorithm increase as D increases. This is reasonable from the introduction of the method. Following numerous simulation experiments, we find that the appropriate number of nodes for the IBA in this article is 15-20.

From Fig. 8, the IBA algorithm can plan a feasible, safe, and effective flight path for UAVs in a three-dimensional environment that can effectively avoid no-fly areas and mountains. It also can be found that the IBA has good convergence from Fig. 5-7.

In general, from the statistical results in Table 2-8, we can find that the population size, the ratio of employed bees to onlooker bees, the number of nodes D, the initial loudness A^0, and α, have a greater impact on the simulation results of the IBA algorithm, however, T_{limit}, F, r^0, and γ have a little effect on the results of the IBA algorithm. It can be seen from the comparison results of IBA with standard BA and ABC that IBA presents better advantages. The convergence speed of IBA is about 50% faster than classical BA, and the quality of the optimum solution is about 40% higher than that of ABC. In addition, compared to traditional swarm intelligence algorithms and improved intelligence algorithms, the optimum solution of IBA is better than them. In other words, the IBA can better solve the UAV flight path-planning problem.

B. FUNCTION OPTIMIZATION PROBLEM

This section mainly verifies the performance of the IBA algorithm on continuous problems. We use four benchmark functions to test the accuracy and convergence of the IBA algorithm and compare it with other swarm intelligence algorithms. The goal of optimization is to minimize the test results of all benchmark functions. Moreover, we run each algorithm 20 times for significant statistical analysis.

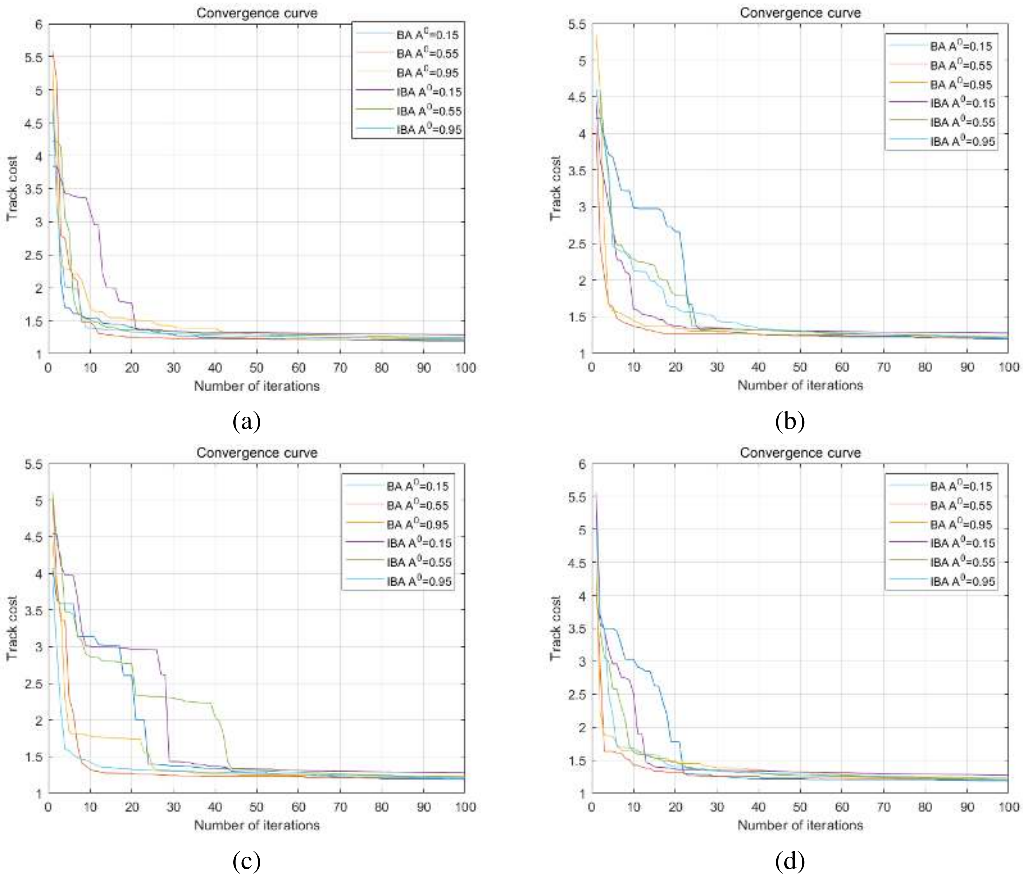


FIGURE 6. The optimal solution convergence curve of BA algorithm and IBA algorithm with different A^0 and α. (a) $\alpha = 0.3$; (b) $\alpha = 0.5$; (c) $\alpha = 0.7$; (d) $\alpha = 0.9$.

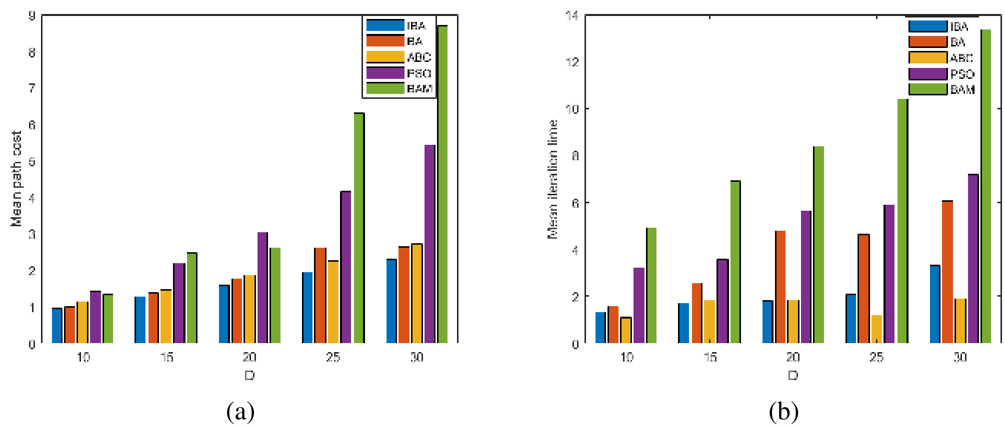


FIGURE 7. Histogram of the average path cost and average iteration time of different algorithms. (a) Average path cost; (b) Average iteration time.

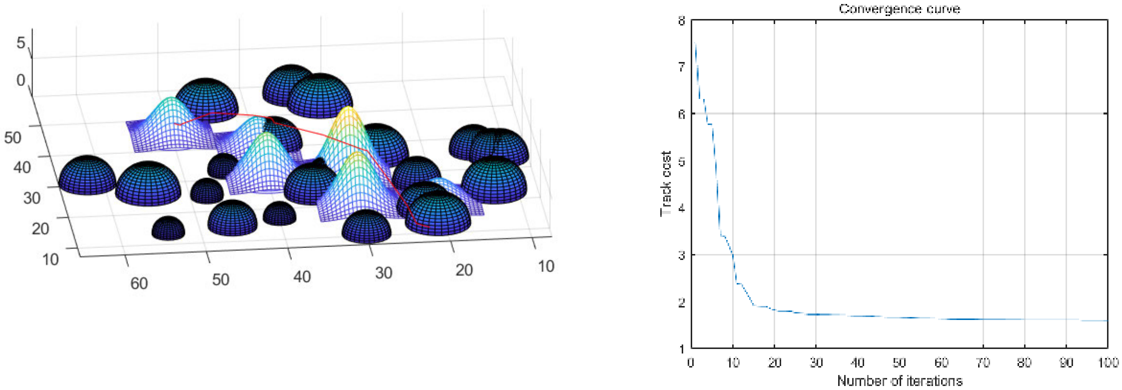
There are many standard test functions for validating new algorithms. In this article, we choose the well-known Rosenbrock's function [31]

$$f_1(\mathbf{x}) = \sum_{i=1}^{d-1}(1 - x_i^2)^2 + 100(x_{i+1} - x_i^2)^2 \tag{16}$$

and De Jong's standard sphere function

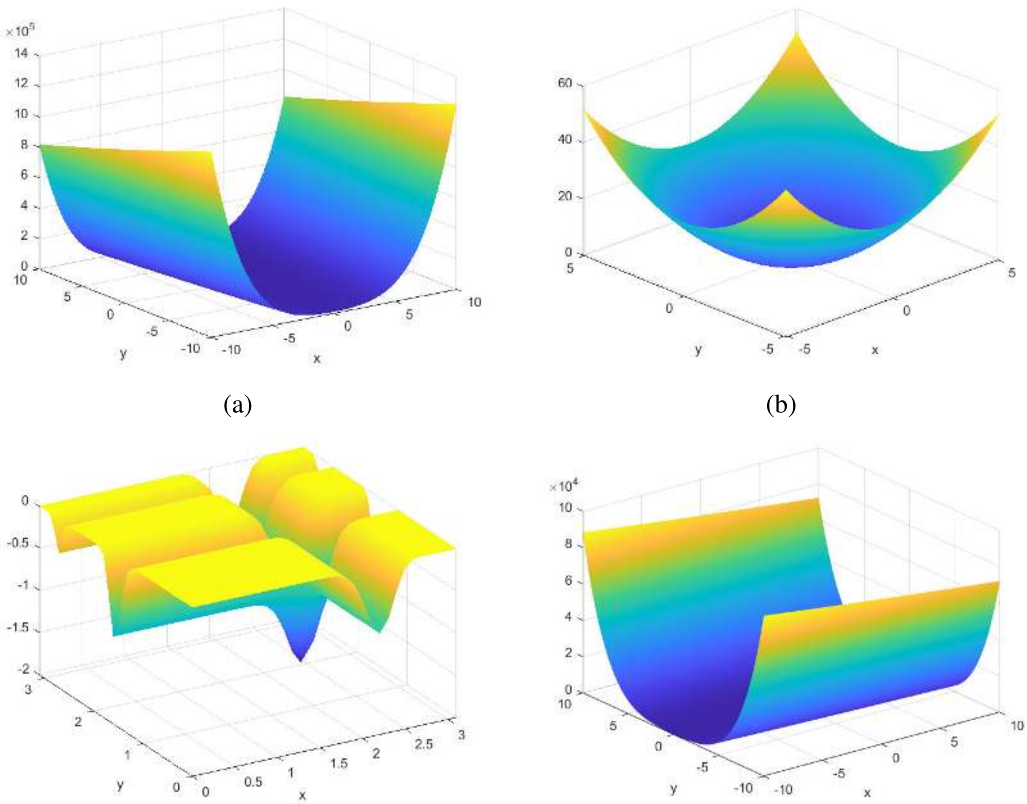
$$f_2(\mathbf{x}) = \sum_{i=1}^{d} x_i^2 \tag{17}$$

We know that $f_1(\mathbf{x})$ has a global minimum $f_1^{min} = 0$ at $(1, 1)$ in 2D and Minimum of $f_2(\mathbf{x})$ is $f_2^{min} = 0$ at $(0, 0, \ldots, 0)$ for any $d \geq 3$.



(a) (b)

FIGURE 8. Path planning of IBA and optimal value convergence curve (a) Path planning; (b) Optimal value convergence curve.



(a) (b)

(c) (d)

FIGURE 9. Four benchmark functions in 3D. (a) Rosenbrock's function; (b)De Jong's standard sphere function; (c) Michalewicz's function; (d) Dixon-Price's function.

Michalewicz's function is also selected to test the algorithm.

$$f_3(\mathbf{x}) = -\sum_{i=1}^{d} sin(x_i)[sin(\frac{ix_i^2}{\pi})]^{2m}. \tag{18}$$

It is usually set m = 10, and the global minimum has been approximated by $f_3^{min} \approx -1.801$ for $d = 2$ and $f_3^{min} \approx -4.687$ for $d = 5$.

In addition, we also added a standard test function, Dixon-Price's function, to perform numerical global optimization, and its global minimum is $f_4^{min} = 0$ for $x_i = 0$,

TABLE 9. Benchmark functions.

Function	Name	Definition
$f_5(\mathbf{x})$	Axis parallel hyper-ellipsoid function	$f_5(\mathbf{x}) = \sum_{i=1}^{d}(ix_i^2)$
$f_6(\mathbf{x})$	Rotated hyper-ellipsoid function	$f_6(\mathbf{x}) = \sum_{i=1}^{d}\sum_{j=1}^{i} x_j^2$
$f_7(\mathbf{x})$	Rastrigin's function	$f_7(\mathbf{x}) = 10d + \sum_{i=1}^{d}[x_i^2 - 10cos(2\pi x_i)]$
$f_8(\mathbf{x})$	SchwefelProblem function	$f_8(\mathbf{x}) = \sum_{i=1}^{d}\lvert x_i\rvert + \prod_{i=1}^{d}\lvert x_i\rvert$
$f_9(\mathbf{x})$	Sum of different power functions	$f_9(\mathbf{x}) = \sum_{i=1}^{d}\lvert x_i\rvert^{i+1}$
$f_{10}(\mathbf{x})$	Styblinski-Tang function	$f_{10}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)$

TABLE 10. Comparison of IBA with BA, ABC, PSO, IABC.

Function	Parameter	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$f_4(\mathbf{x})$	$f_5(\mathbf{x})$	$f_6(\mathbf{x})$	$f_7(\mathbf{x})$	$f_8(\mathbf{x})$	$f_9(\mathbf{x})$	$f_{10}(\mathbf{x})$	Average
IBA	Mean	**141.9618**	**3.15E-06**	**-49.9295**	**2.54**	**5.77E-05**	**6.47E-05**	**0.0017**	**0.0205**	**2.28E-07**	**-1958.3**	**-186.3700774**
	Standard deviation	**1.71E+02**	**4.64E-06**	**0.0297**	**2.67**	**3.55E-05**	**9.09E-05**	**0.0013**	0.0044	**4.35E-07**	**1.61E-04**	**17.33346921**
	Number of changes	658950	419196	421790	419125	419134	419084	419179	437971	419132	419006	421165.7
	Number of comparisons	1.10E+06	1.10E+06	1.08E+06	1.10E+06	1.10E+06	1.10E+06	1.10E+06	1.11E+06	1.09E+06	1.09E+06	1052605
ABC	Mean	1.35E+03	2.17E-05	-48.9209	5.22	6.11E-04	5.50E-04	0.0035	0.022	1.61E-05	-1958.3	-64.8870201
	Standard deviation	5.48E+03	7.37E-06	0.1262	3.14	3.48E-04	3.26E-04	0.0032	0.0034	2.15E-05	1.75E-04	548.4569378
	Number of changes	299973.3	299975.85	299969.05	299985.45	299971.45	299977.05	299977.25	299972.8	299973	299968.95	299974.415
	Number of comparisons	4.58E+05	4.58E+05	3.19E+05	4.58E+05	4.58E+05	4.58E+05	4.52E+05	4.57E+05	4.42E+05	4.52E+05	441068
BA	Mean	4.02E+03	12.6316	-10.6869	6.73E+03	452.7984	418.057	562.8053	6.58E+65	6.18E+06	-1425	6.5815E+64
	Standard deviation	3.00E+03	0.7311	0.7386	5.46E+03	122.4133	101.6821	88.849	2.89E+66	1.69E+07	60.2772	2.8861E+65
	Number of changes	4.50E+05	4.50E+05	5.70E+05	4.50E+05	4.50E+05	4.50E+05	450054	5.64E+05	4.50E+05	450017	473435.1
	Number of comparisons	1.51E+05	1.51E+05	2.70E+05	1.51E+05	1.52E+05	151535	1.51E+05	2.64E+05	1.52E+05	1.51E+05	174644.5
IABC	Mean	751.0559	2.07E-05	-48.9315	4.9006	5.34E-04	4.98E-04	0.008	0.0214	1.51E-05	-1958.3	-125.1244532
	Standard deviation	2.32E+03	8.63E-06	0.141	3.0114	2.54E-04	2.12E-04	0.0029	0.0033	2.70E-05	2.22E-04	232.6659324
	Number of changes	305492.05	305499.8	305758.85	305497.5	305505.4	305503.5	305527.1	305508.9	315558	305515.9	306536.7
	Number of comparisons	4.61E+05	4.61E+05	3.22E+05	4.61E+05	4.61E+05	4.61E+05	4.61E+05	4.60E+05	4.59E+05	4.55E+05	446124
PSO	Mean	1.06E+03	3.60E-03	-8.5379	6.5706	0.14	0.1558	164.4823	8.32E+16	1.50E-05	-1618.1	8.3211E+15
	Standard deviation	3.10E+03	7.26E-04	-9.5457	4.6188	2.94E-02	0.0506	27.4828	3.72E+17	6.23E-06	50.1339	3.721E+16
	Number of changes	299950	299950	299950	299950	299950	299950	299950	299950	299950	299950	299950
	Number of comparisons	945	1.81E+04	416.5	1.21E+04	2.16E+04	2.14E+04	9.63E+03	3.61E+03	2.19E+04	9.86E+03	11952.22

$i = 1, 2, \ldots, d$.

$$f_4(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2. \tag{19}$$

However, the minimum value of Michalewicz's function can be less than 0, so we have modified the fitness of the IBA algorithm by Eq 20.

$$fit(x_i) = \begin{cases} \dfrac{1}{1+f(x_i)}, & f(x_i) > 0 \\ 1 + \lvert f(x_i)\rvert, & \text{otherwise.} \end{cases} \tag{20}$$

where $f(x_i)$ is the function value of x_i.

Moreover, we add other benchmark functions in Table 9, and set the initial x_i, $i = 1, 2, \ldots, N_p$ range of each benchmark function to $[-100, 100]$.

We reset the maximum number of evaluations to 3×10^5, dimension $d = 50$, $F \in [-1, 1]$ and other parameters are the same as the previous simulation experiment.

From the statistical results in Table 10, we can see that for each benchmark function, IBA can obtain a better quality solution and a lower standard deviation, compared to BA, ABC, PSO, and IABC. From the average of these 10 benchmark functions, IBA can also ensure that the obtained mean and standard deviation are the smallest. We believe that it is worthwhile to perform multiple changes and comparisons for individuals to obtain better solutions. Obviously,in the process of finding the global minimum of different functions, IBA is more suitable for finding the global minimum solution within a fixed number of evaluations.

VI. CONCLUSION

The purpose of this paper is to make the UAV obtain a crash-free, safer, and shorter flight path. An improved bat algorithm (IBA) is proposed that integrates elements of the ABC, employed bee, onlooker bees, and scout bees. In the IBA, the employed bee searches for the path according to the behavior of bats using sonar positioning. To increase the local search ability, a mutation factor is considered. When the individual falls into a local optimum, the scout bee searches for a new path to replace the old one. In addition, this work also proves in detail that the IBA is convergent and solves the function optimization problem to prove that the IBA algorithm has the potential for broad application.

We tested all the setting parameters in the IBA in this paper. Based on the statistical results, it can be found that when T_{limit}, F, r^0, and γ are changed, the average path cost and iteration time of IBA do not change significantly, that is to say, T_{limit}, F, r^0, and γ have a little impact on the performance of the IBA. In other aspects, A^0 and α mainly affect the updating of information from the bat algorithm part of the IBA, and the larger A^0 and α, the better the result obtained by the IBA. Similarly, the ratio of employed bees to onlooker bees also affects the results of the IBA. This is mainly due to the different behaviors of employed bees and onlooker bees in the search paths. The increase in population size can inevitably increase the iteration time of the algorithm. The increase in D increases the path cost mainly because the method of calculating the cost of a path is flawed, and the path cost cannot be calculated accurately. From a statistical data point of view, the ratio of employed bees to onlooker bees, population size, and D have greater influence than A^0 and α.

In this article, a large number of simulation experiments have confirmed that the IBA can quickly plan a flight path for the UAVs, effectively avoiding mountains and various threatening no-fly areas. Additionally, based on the comparison results, the IBA is superior to DE, BAM, ABC, PSO, BA, BA-ABC, IABC, and GFACO in the battlefield environment of this article. The convergence speed of the IBA is about 50% lower than the standard BA. Moreover, compared to the ABC algorithm, the IBA sacrifices very little convergence time to improve the quality of the optimal solution by about 14%. In terms of function optimization, compared with ABC, BA, IABC, and PSO, IBA can get higher quality solutions, lower standard deviation. On the other hand, from the perspective of time complexity, suppose the number of populations is n, the problem solution is D-dimensional, and the ratio of employed bees to onlooker bees is Ra. After t iterations, the time complexity of the IBA is approximately $O(nD+t(Ra\cdot nD+(1-Ra)n))$, which is less than the time complexity of the BA ($O(nD+tnD)$) and this is consistent with the experimental results. In future work, we will study the use of the IBA to solve the UAV flight path-planning problem in a dynamic environment.

APPENDIX
IBA ALGORITHM CONVERGENCE

This chapter mainly proves the convergence of IBA through Markov chain. In order to illustrate the Markov chain model of IBA, we need to give some related mathematical descriptions and definitions.

Definition 1 [42]: State equivalence. Suppose $S = \{s = (X_1, X_2, \ldots, X_{N_P}) | X_i \in \mathbf{X}, 1 \le i \le N_P\}$ is the state space of the group which is composed of the set of all possible states of the group. $\mathbf{X} = \{X | X \in A\}$ is the individual state space, and is the feasible solution space. $\varphi(s, X) = \sum \chi_{|X|(X_i)}$ is the number of group state contains individual state X, where $\chi_{|B|}$is the indicative function of set B, $s \in S$, $X \in s$. If $\exists s_1, s_2 \in S$, such that $\forall X \in \mathbf{X}$, there is $\varphi(s_1, X) = \varphi(s_2, X)$.

Definition 2 [43]: The group state equivalence class induced by the equivalence relation "$\sim$" on S is denoted as $L = S/\sim$. The equivalence class of this group has the following properties:

(1). Any group state in a certain equivalence class L is equivalent, that is, $s_1 \sim s_2, \forall s_1, s_2 \in L$.

(2). The state of any group within L is not equivalent to the state of any group outside L, that is, $s_1 \nsim s_2$, $\forall s_1 \in L$, $s_2 \in L$.

(3). Any two different equivalence classes have no intersection, that is, $L_1 \cap L_2 = \varnothing, \forall L_1 \ne L_2$.

Definition 3: Group state transition. $T_X(X_i) = X_j$ is the group state one-step transition from state X_i to state X_j, where $X_i, X_j \in X$.

After that, we discuss the state transition probability of the IBA. According to the structure of the IBA, we can get the one-step transition probability of the individual state from X_i to X_j.

$$p(T_X(X_i) = X_j) = \begin{cases} p_{ba}(T_X(X_i) = X_j), \\ \text{realized by bats} \\ p_{on}(T_X(X_i) = X_j), \\ \text{realized by onlooker bees} \\ p_{sc}(T_X(X_i) = X_j), \\ \text{realized by scout bees} \\ p_{ba}(T_X(X_i) = X_j) \times p_{on}(T_X(X_i) = X_j), \\ \text{realized by bats and onlooker bees.} \end{cases}$$

Without considering the population number and dimension in Eq.8, we can get

$$V_t = V_{t-1} + (X_{t-1} - P_g)f_i \tag{21}$$

Thereby, Eq.22 is established.

$$X_t = (2 + f_i)X_{t-1} - X_{t-2} - P_g f_i \tag{22}$$

where P_g is global optimal position.

According to Definition 3 and the geometric properties of the IBA, the one-step transition probability of a bat from state X_i to state X_j is:

$$\begin{aligned} &p_{ba}(T_X(X_i) = X_j) \\ &\quad = p_{ba}(V_i \to V_j) \times p_{ba}(X_i \to X_j) p_{ba}(P_{g_i} \to P_{g_j}) \end{aligned} \tag{23}$$

Through individual iterative formula and location update criterion, we can get Eq.24

$$p_{ba}(P_{g_i} \to P_{g_j}) = \begin{cases} 1, & f(P_{g_i}) \le f(P_{g_j}) \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

Assuming that the individual has n dimensions. Then the one-step transition probability of speed and position is:

$$\begin{aligned} &p_{ba}(V_i \to V_j) \\ &= \begin{cases} \frac{1}{|\triangle_1|}, & V_j \in [V_i, V_i + f_i(X_i - P_{g_i})] \\ 1, & j = i+1 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{25}$$

$p_{ba}(X_i \to X_j)$

$$= \begin{cases} \frac{1}{|\triangle_2|}, & X_j \in [V_i + X_i, V_i+X_i+f_i(X_i - P_{g_i})] \\ & and\ rand_1 < r_i\ and\ rand_2 < A_i \\ & and\ f(X_i) < f_{X_j} \\ \frac{1}{|P_{g_i}+\varepsilon \cdot rand|}, & X_j \in [P_{g_i} - \varepsilon \cdot rand, P_{g_i}+\varepsilon \cdot rand] \\ & and\ rand_1 > r_i\ and\ rand_2 < A_i \\ & and\ f(X_i) < f_{X_j} \\ 1, & j = i+1 \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

where $|\triangle_1| = \int_{v_{i_1}}^{v_{j_1}} \int_{v_{i_2}}^{v_{j_2}} \cdots \int_{v_{i_n}}^{v_{j_n}} dv_n \cdots dv_2 dv_1$, $|\triangle_2| = \int_{x_{i_1}}^{x_{j_1}} \int_{x_{i_2}}^{x_{j_2}} \cdots \int_{x_{i_n}}^{x_{j_n}} dx_n \cdots dx_2 dx_1$

Similarly, we you can get

$$p_{on}(T_s(X_i) = X_j) = \begin{cases} \frac{1}{|X_i - X_j|} p_1(X_i \to X_j), & X_j \in [X_i - (X_i - X_k), \\ & X_i + (X_i - X_k)] \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

$$p_{sc}(T_s(X_i) = X_j) = \begin{cases} \frac{1}{|X_{max} - X_{min}|}, & X_j \in [X_{min}, X_{max}] \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

where in Eq.21-28, X is multi-dimensional data, X_k is a randomly selected solution within the range of feasible solutions.

$$p_1(X_i \to X_j) = \begin{cases} 1, & f(X_i) < f(X_j) \\ 0, & \text{otherwise.} \end{cases}$$

Definition 4 [42]: For $\forall s_i, s_j \in S$, in the IBA, the individual state is one-step transferred from state s_i to state S_j, denoted as $T_S(s_i) = s_j$. The one-step transition probability for all individual states in-group s_i to simultaneously transfer to all individual states in group s_j is:

$$p(T_S(s_i = s_j)) = \prod_{m=1}^{N_p} p(T_S(X_{i_m}) = X_{j_m}) \tag{29}$$

Definition 5 [42]: Assume that $L_i = (s_{i_1}, s_{i_2}, \ldots, s_{i_n})$ and $L_j = (s_{j_1}, s_{j_2}, \ldots, s_{j_n})$ are the equivalence classes of any two group states caused by the equivalence relation "$\sim$" on S. L_i one-step transfers to L_j, denoted as $T_L(L_i) = L_j$, then the one-step transition probability of $T_L(L_i) = L_j$ is:

$$p(T_L(L_i) = L_j) = \sum_{a=1}^{n} \sum_{b=1}^{m} p(T_S(s_{i_a}) = s_{j_b}) \tag{30}$$

Definition 6 [44]: Assuming that the optimal solution of optimization problem $\langle A, f \rangle$ is g^*, define the group's optimal state set $G = \{s^* = (X) | f(X) = f(g^*), s \in S\}$.

If $G = S$, then every solution in the feasible solution space is the optimal solution and feasible solution, and optimization is meaningless at this time. So we discuss the convergence of the IBA algorithm in the case of $G \subset S$.

REFERENCES

[1] Z. H. Hu, "Research on Some key techniques of UAV path planning based on intelligent optimization algorithm," Ph.D. dissertation, Dept. Auto. Eng., NUAA Univ., Nanjing, China, 2011.

[2] Y. Meng, J. Su, J. Song, W.-H. Chen, and Y. Lan, "Experimental evaluation of UAV spraying for peach trees of different shapes: Effects of operational parameters on droplet distribution," *Comput. Electron. Agricult.*, vol. 170, Mar. 2020, Art. no. 105282, doi: 10.1016/j.compag.2020.105282.

[3] K. Yang, G. Y. Yang, and S. I. Huang Fu, "Research of control system for plant protection UAV based on pixhawk," *Procedia Comput. Sci.*, vol. 166, pp. 371–375, Jan. 2020, doi: 10.1016/j.procs.2020.02.082.

[4] P. Yao, H. Wang, and Z. Su, "UAV feasible path planning based on disturbed fluid and trajectory propagation," *Chin. J. Aeronaut.*, vol. 28, no. 4, pp. 1163–1177, Aug. 2015, doi: 10.1016/j.cja.2015.06.014.

[5] Y. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhanov, "On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget," *Sci. Rep.*, vol. 8, no. 1, pp. 1–9, Jan. 2018, doi: 10.1038/s41598-017-18940-4.

[6] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired Metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377, doi: 10.1016/j.eswa.2020.113377.

[7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[8] Z. Wang, L. Liu, T. Long, and Y. Wen, "Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposition-based genetic algorithm with double-chromosome encoding," *Chin. J. Aeronaut.*, vol. 31, no. 2, pp. 339–350, Feb. 2018, doi: 10.1016/j.cja.2017.09.005.

[9] C. R. Atencia, J. Del Ser, and D. Camacho, "Weighted strategies to guide a multi-objective evolutionary algorithm for multi-UAV mission planning," *Swarm Evol. Comput.*, vol. 44, pp. 480–495, Feb. 2019, doi: 10.1016/j.swevo.2018.06.005.

[10] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008, doi: 10.1109/TEVC.2008.919004.

[11] L. P. Behnck, D. Doering, C. E. Pereira, and A. Rettberg, "A modified simulated annealing algorithm for SUAVs path planning," *IFAC-PapersOnLine*, vol. 48, no. 10, pp. 63–68, 2015, doi: 10.1016/j.ifacol.2015.08.109.

[12] R. A. Formato, "Central force optimization: A new deterministic gradient-like optimization Metaheuristic," *Opsearch*, vol. 46, no. 1, pp. 25–51, Mar. 2009, doi: 10.1007/s12597-009-0003-4.

[13] R. A. Formato, "Central force optimization with variable initial probes and adaptive decision space," *Appl. Math. Comput.*, vol. 217, no. 21, pp. 8866–8872, Jul. 2011, doi: 10.1016/j.amc.2011.03.151.

[14] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007, doi: 10.1007/s10898-007-9149-x.

[15] J. Karimi and S. H. Pourtakdoust, "Optimal maneuver-based motion planning over terrain and threats using a dynamic hybrid PSO algorithm," *Aerosp. Sci. Technol.*, vol. 26, no. 1, pp. 60–71, Apr. 2013, doi: 10.1016/j.ast.2012.02.014.

[16] C. Pan, H. Wang, J. Li, and M. Korovkin, "Path planning of mobile robot based on an improved ant colony algorithm," presented at the CCIS, Najing, China, 2018, doi: 10.1007/978-3-030-374365-11.

[17] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generat. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, Jun. 2000, doi: 10.1016/S0167-739X(00)00043-1.

[18] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "A bat algorithm with mutation for UCAV path planning," *Sci. World J.*, vol. 2012, pp. 1–15, Nov. 2012, doi: 10.1100/2012/418946.

[19] C. Sur and A. Shukla, "Adaptive & discrete real bat algorithms for route search optimization of graph based road network," presented at the IEEE Int. Conf. Mach. Intell. Res. Adv., Katra, India, Dec. 2013.

[20] Y. Wang, K. Li, Y. Han, F. Ge, W. Xu, and L. Liu, "Tracking a dynamic invading target by UAV in oilfield inspection via an improved bat algorithm," *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106150, doi: 10.1016/j.asoc.2020.106150.

[21] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[22] A. A. Najm and I. K. Ibraheem, "Nonlinear PID controller design for a 6-DOF UAV quadrotor system," *Eng. Sci. Technol., Int. J.*, vol. 22, no. 4, pp. 1087–1097, Aug. 2019, doi: 10.1016/j.jestch.2019.02.005.

[23] A. A. Najm and I. K. Ibraheem, "Altitude and attitude stabilization of UAV quadrotor system using improved active disturbance rejection control," *Arabian J. Sci. Eng.*, vol. 45, no. 3, pp. 1985–1999, Mar. 2020, doi: 10.1007/s13369-020-04355-3.

[24] I. K. Ibraheem, "Anti-disturbance compensator design for unmanned aerial vehicle," *J. Eng.*, vol. 26, no. 1, pp. 86–103, Dec. 2019, doi: 10.31026/j.eng.2020.01.08.

[25] W. R. Abdul-Adheem, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Improved active disturbance rejection-based decentralized control for MIMO nonlinear systems: Comparison with the decoupled control scheme," *Appl. Sci.*, vol. 7, no. 10, pp. 1–29, Mar. 2020, doi: 10.3390/app10072515.

[26] A. A. Najm, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Genetic optimization-based consensus control of multi-agent 6-DoF UAV system," *Sensors*, vol. 12, no. 20, pp. 86–103, Jun. 2020, doi: 10.3390/s20123576.

[27] A. J. Humaidi, I. K. Ibraheem, A. T. Azar, and M. E. Sadiq, "A new adaptive synergetic control design for single link robot arm actuated by pneumatic muscles," *Entropy*, vol. 7, no. 22, pp. 1–24, Jun. 2020, doi: 10.3390/e22070723.

[28] W. R. Abdul-Adheem, I. K. Ibraheem, A. J. Humaidi, and A. T. Azar, "Model-free active input–output feedback linearization of a single-link flexible joint manipulator: An improved active disturbance rejection control approach," *Meas. Control*, Jun. 2020, Art. no. 002029402091717. [Online]. Available: https://journals.sagepub.com/doi/10.1177/ and in part by the Fundamental Research Funds for the Central Universities under Grant 2018IB0170020294020917171, doi: 10.1177/0020294020917171.

[29] I. K. Ibraheem and F. H. Ajeil, "Path planning of an autonomous mobile robot using swarm based optimization techniques," *Al-Khwarizmi Eng. J.*, vol. 12, no. 4, pp. 12–25, Dec. 2017, doi: 10.22153/kej.2016.08.002.

[30] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments," *Sensors*, vol. 7, no. 20, pp. 1–26, Mar. 2020, doi: 10.3390/s20071880.

[31] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization* (Studies in Computational Intelligence), vol. 284, J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-642-12538-6_6.

[32] A. Chakri, R. Khelif, M. Benouaret, and X.-S. Yang, "New directional bat algorithm for continuous optimization problems," *Expert Syst. Appl.*, vol. 69, pp. 159–175, Mar. 2017, doi: 10.1016/j.eswa.2016.10.050.

[33] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *J. Comput. Sci.*, vol. 5, no. 2, pp. 224–232, Mar. 2014, doi: 10.1016/j.jocs.2013.10.002.

[34] T. T. Nguyen, J. S. Pan, T. K. Dao, M. Y. Kuo, and M. F. Horng, "Hybrid bat algorithm with artificial bee colony," in *Intelligent Data Analysis and Its Applications, Volume II* (Advances in Intelligent Systems and Computing), vol. 298, J. S. Pan, V. Snasel, E. Corchado, A. Abraham, and S. L. Wang, Eds. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-07773-4_5.

[35] I. K. Ibraheem, F. H. Ajeil, and Z. H. Khan, "Path planning of an autonomous mobile robot in a dynamic environment using modified bat swarm optimization," 2018, *arXiv:1807.05352*. [Online]. Available: https://arxiv.org/abs/1807.05352

[36] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 3, May 2020, Art. no. 172988142092949, doi: 10.1177/1729881420929498.

[37] N. Lin, J. Tang, X. Li, and L. Zhao, "A novel improved bat algorithm in UAV path planning," *Comput., Mater. Continua*, vol. 61, no. 1, pp. 323–344, 2019, doi: 10.32604/cmc.2019.05674.

[38] C. Xu, H. Duan, and F. Liu, "Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning," *Aerosp. Sci. Technol.*, vol. 14, no. 8, pp. 535–541, Dec. 2010, doi: 10.1016/j.ast.2010.04.008.

[39] G.-G. Wang, H. E. Chu, and S. Mirjalili, "Three-dimensional path planning for UCAV using an improved bat algorithm," *Aerosp. Sci. Technol.*, vol. 49, pp. 231–238, Feb. 2016, doi: 10.1016/j.ast.2015.11.040.

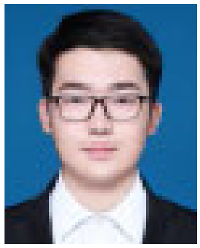
[40] S. Aslan and S. Demirci, "Solving UAV localization problem with artificial bee colony (ABC) algorithm," in *Proc. 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2019, pp. 735–738, doi: 10.1109/UBMK.2019.8907034.

[41] L. Lei and Q. Shiru, "Path planning for Unmanned Air Vehicles using an improved artificial bee colony algorithm," presented at the 31st Chin. Control Decis. Conf., Hefei, China, 2012.

[42] R. Zi-hui, W. Jian, and G. Yue-lin, "The global convergence analysis of particle swarm optimization algorithm based on Markov chain," *Control Theory Appl.*, vol. 28, no. 4, pp. 462–466, Apr. 2011.

[43] N. Ai-ping and Z. Xue-ying, "Convergence analysis of artificial bee colony algorithm," *Control Decis.*, vol. 28, no. 10, pp. 1554–1558, Oct. 2013.

[44] S. Junna, C. Tao, Y. Keqiang, and L. Shen, "Markov chain model analysis of bat algorithm," *Comput. Eng.*, vol. 43, no. 7, pp. 198–202, Jul. 2017.



XIANJIN ZHOU was born in 1997. He received the bachelor's degree in mathematics and applied mathematics from Anhui Normal University, in 2019. He is currently pursuing the M.S. degree with the Department of Mathematics, Wuhan University of Technology. His research interests include path planning of UAV and intelligence optimization algorithm.

FEI GAO received the B.S. degree in normal mathematics and the M.S. degree in applied mathematics from Wuhan University, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in fluid mechanics from the Wuhan University of Technology, Wuhan, China, in 2006. From 2008 to 2009, he held a postdoctoral position with the Future Of Beyond Human Intelligence Lab, Department of Electrical Engineering Computer Science, Korea Advanced Institute of Science and Technology (KAIST). From 2011 to 2012, he was a Marie-Curie Fellow with the Department of Electronics and Telecommunications, Norges teknisknaturvitenskapelige universitet (NTNU). He is currently a Professor with the Department of Mathematics, School of Science, Wuhan University of Technology. His current research interests include swarm intelligence (especially particle swarm optimizations, artificial bee colony algorithm, bacterial chemotaxis optimization algorithm, and ant colony optimizations), bioinformatics, intelligence optimization methods, evolutionary computation (differential evolution algorithm, genetic algorithms etc.), quantum intelligence computation, global optimization, and non-Lyapunov methods in chaos control, chaos synchronization, chaotic characters.

XI FANG was born in 1981. He received the Ph.D. degree from the Wuhan University of Technology, Wuhan, China, in 2011. He was an Associate Professor. His current research interests are swarm intelligence, mathematical modelling in engineering. He has published more than 12 journal articles as the major author, among them, five articles were retrieved by SCI/EI.

ZEHONG LAN was born in Zhongxiang, Hubei, China. He is currently pursuing the B.S. degree in mathematical science with Wuhan University, Wuhan, China. His research interests include computational mathematics, artificial intelligence, and topology.