*Robotica* (2023), **41**, pp. 1244–1257
doi:10.1017/S0263574722001680

RESEARCH ARTICLE

# Multi-UAV path planning based on IB-ABC with restricted planned arrival sequence

Li Tan[1], Jiaqi Shi[1], Jing Gao[1,*], Haoyu Wang[1], Hongtao Zhang[1] and Yu Zhang[2]

[1]School of Computer Science and Engineering, Beijing Technology and Business University, Beijing 100048, China and
[2]School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
*Corresponding author. E-mail: gaojing@btbu.edu.cn

**Received:** 13 October 2021; **Revised:** 24 October 2022; **Accepted:** 2 November 2022;
**First published online:** 12 December 2022

**Keywords:** ABC algorithm, multi-UAV path planning, feedback, element point control, balance strategy

## Abstract

Path planning is a key research issue in the field of unmanned aerial vehicle (UAV) applications. In practical applications, multi-objective path planning is usually required for multi-UAVs, so this paper proposes the improved balanced artificial bee colony (IB-ABC) algorithm to optimize multi-objective path planning. The algorithm adopts the ABC algorithm that emphasizes the global search capability, which is based on iterative feedback information. It uses single-element points, multi-element points, and iteration constraints to optimize the strategies of employed bees, follower bees, and scout bees, respectively. In terms of time and priority, simulation experiments prove that the IB-ABC algorithm can balance local and global search capabilities, accelerate the speed of convergence, and realize multi-UAV path planning in complex mountain environments.

## 1. Introduction

The main goal of the unmanned aerial vehicle (UAV) path planning problem is to find a feasible path, which requires the UAV to reach the destination safely without encountering any collisions [1]. In recent years, with the increase in popularity of UAV application fields, a single UAV has gradually become difficult to meet the requirements of different applications. This has led to the research on UAV path planning to expand from single UAV to multiple UAVs [2]. In the UAV path planning research, multiple factors such as path length, path safety, and path concealment need to be considered. Therefore, multiple optimization goals need to be considered at the same time to obtain an accurate solution. Multi-UAV path planning can be regarded as a multi-objective optimization problem (MOOP) [3].

Among the traditional algorithms to solve the MOOP problem of path planning, the heuristic optimization algorithm has attracted the attention of researchers due to its high convergence efficiency [4]. In particular, evolutionary algorithms such as the genetic algorithm (GA), the ant colony algorithm (ACO), and the particle swarm algorithm (PSO) have been widely used in various fields due to their robustness and universal applicability. However, the aforementioned algorithms have been widely used due to their strong local search capabilities. It is easy to fall into the local optimum [5].

In GA, a small number of individuals with better fitness function values appear in the evolution process, and their fitness function values are constantly approaching the average fitness value. The proportion of the population in the population is increasing. The mutation operator cannot make the algorithm proceed further and produce the local optimum. In ACO, when the number of ants is small and the scale of the problem is large, the pheromone on the path that has never been searched is reduced to close to 0, the randomness of the search is weakened, and the local optimum is prone to appear. In PSO, due to the nature of the function to be optimized and the rapid disappearance of the diversity of particles in the PSO, premature convergence is caused, and PSO will fall into a local optimum. The artificial bee colony (ABC) algorithm does not need to know the special information of the problem. It only needs to compare the pros and cons of the problem. Through the local optimization behaviors of individual artificial bees, the global optimal value is finally highlighted in the group, and it has better global optimization capabilities.

The ABC algorithm is proposed by Karaboga. It is known for its multi-variable global optimization capability, which can overcome the trap of local optimization while searching for the global [6]. In order to make better use of the advantages of the ABC algorithm, Chiang *et al*. proposed a discretized optimized honey source algorithm, which is adjusted together with the core parameters of the support vector machine to enhance the classification accuracy and convergence speed. However, this reduces the local search ability to some extent [7]. Local search strategy in neighborhood space by Hussain *et al*. and Chaos theory by Alatas use different search strategies to strengthen the local search ability of the ABC algorithm [8, 9]. Deniz *et al*. combined the differential evolution algorithm and the ABC algorithm for research, which reduced the risk of falling into the local optimum with a greater probability in the early search stage [10]. Li *et al*. used a combination of roulette and anti-roulette strategies to obtain the initial result through interpolation, which improved the algorithm's ability to jump out of the local optimum to a certain extent [11]. The aforementioned methods all improve the local search ability by optimizing a certain strategy in the ABC algorithm but do not pay attention to the imbalance problem in the global search ability of the ABC algorithm.

In the study of path planning for robots, some methods of operations research are also used. For example, literature [12] proposes a method for path planning using operations research tools, and literature [13] proposes an adaptive mutation GA.

In summary, due to the advantages of the ABC algorithm in its global search ability, most studies only focus on improving its local search ability or amplifying the single level of the global search ability without considering the balanced search of the algorithm [12, 13]. In response to this situation, this paper proposes the improved balanced (IB)-ABC algorithm. By introducing iterative feedback information, single-element point control, and double-element point control, the ABC algorithm's strategies for employed bees, follower bees, and scout bees are optimized. Furthermore, the local search and detection strategies are optimized. The algorithm balances local search and global search capabilities, that is, the algorithm has both jumped out of local minimal function and a global optimal function.

## 2. Mathematical modeling of path planning

### 2.1. Environment modeling

Environmental modeling is the prerequisite for the simulation of drones to perform various operations. In our experiment, the grid graph method is used to model the environment and the working environment of the UAV is divided into a series of grid areas of identical size. The schematic diagram of environmental modeling is shown in Fig. 1.

### 2.2. Path initialization using a balanced split strategy

In ABC-related algorithms, the employed bee and follower bee strategies are local optimization strategies, whereas the scout bee strategy is a global optimization strategy [14]. Each nectar represents a feasible solution of the optimization problem, which is a feasible path composed of an array of path codes in the path planning problem [15]. The honey source-optimized UAV flight path corresponding to the employed bee, the follower bee, and the scout bee are optimized solutions obtained by three strategies.

In the three-dimensional rectangular coordinate system, each path to be optimized is coded to generate an array. The flight path of the drone is represented by the coordinate points composed of three integers of X, Y, and Z in the coordinate system. The form of the encoding array is a singly linked list of

https://doi.org/10.1017/S0263574722001680 Published online by Cambridge University Press

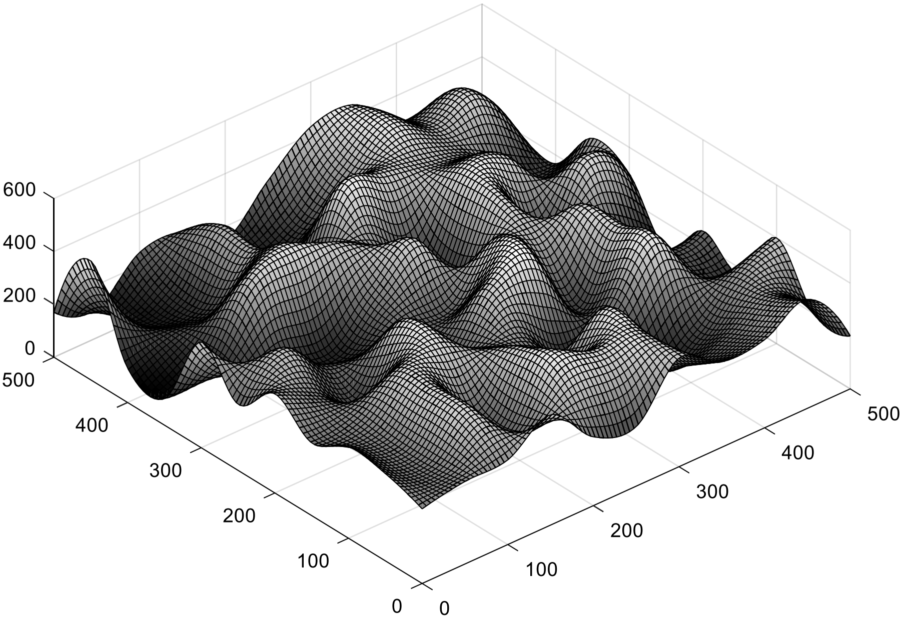


***Figure 1.*** *Schematic diagram of environment modeling.*

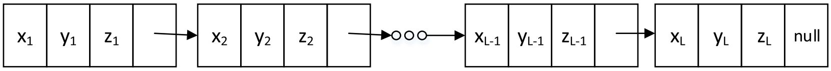


***Figure 2.*** *Path encoding array singly linked list.*

length L. As shown in Fig. 2, $L$ is also the dimension and the number of elements in the path planning IB-ABC algorithm, which is the number of coordinate points in the path. The encoding array is used as the path plan and the nectar source location in the IB-ABC algorithm is used to initialize, and randomly generated SN nectar sources are the SN paths in the path planning IB-ABC algorithm.

The initialization of the path adopts a balanced split strategy, and the process is as follows:

Set $\mathrm{X_i}$ to denote the UAV flight path (original UAV path) corresponding to the i-th nectar source in the three-dimensional Cartesian coordinate system space ($i = 1, 2, \ldots, \mathrm{SN}$), so that the number of nectar sources, employed bees, and follower bees are all equal to SN. Connect the start point $S$ and the end point $T$ in the coordinate system to form a line segment $ST$ and then divide it into $D_l + 1$ parts of equal length with $D_l$ dashed lines perpendicular to $ST$ $\{L_\mathrm{k}\mathrm{k} = 1, 2\ldots, D_1\}$. Starting from each choose any point on $L_\mathrm{k}$ as the coordinate point of the path, where $D_1$ vertical dashed line corresponds to $D$ path points ($D_1$=$D$), connect the $D$ path points on each $L_\mathrm{k}$ in turn to form a feasible path from $S$ to $T$, as shown in Fig. 3. Therefore, the path composed of the coordinates of the corresponding $D$ points is the optimization objective, namely $X_\mathrm{i}$. The specific initialization process of each path point $x_i^j$ in $X_\mathrm{i}$ is shown in Eq. (1).

$$x_i^j = x_{\min}^j + \varphi\left(x_{\max}^j - x_{\min}^j\right) \tag{1}$$

where $x_i^j$ represents the $j$-th element of the i-th feasible solution ($j = 1, 2, \ldots, \mathrm{D}$), that is, the $j$-th coordinate point in the i-th path. $x_{\max}^j$ and $x_{\min}^j$ represent the upper part of the $j$-th coordinate point, represented as a line connecting the square points in Fig. 3. The lower limit D represents the dimension and the number of elements of the feasible solution, that is, the number of coordinate points in the path, and $\varphi$ represents a random number between 0 and 1.

https://doi.org/10.1017/S0263574722001680 Published online by Cambridge University Press

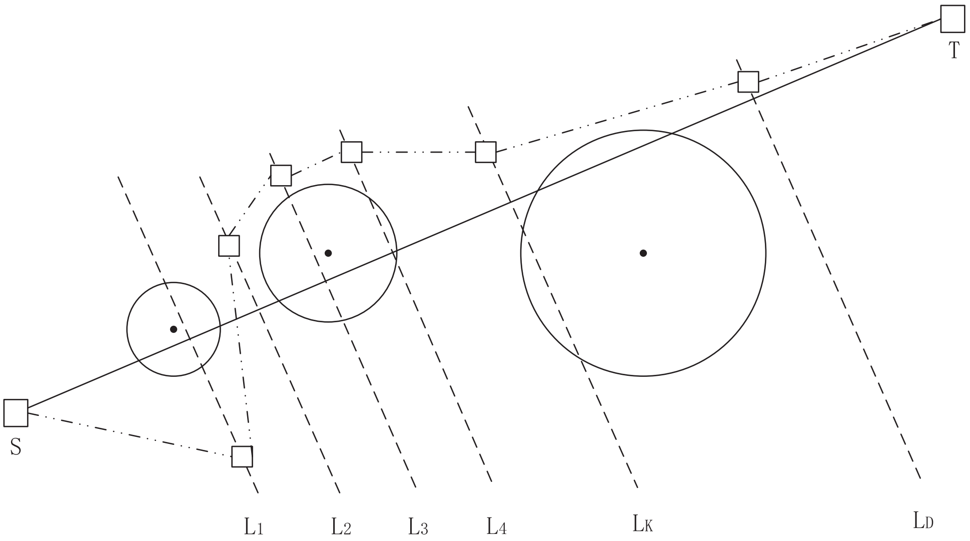


**Figure 3.** *Balanced segmentation strategy.*

### 2.3. Fitness function design

The factors closely related to the UAV path include path length and safety [16, 17]. The fitness function in this paper includes route length, airway smoothness, airway concealment, and airway threat.

#### 2.3.1. Route length

The fitness function aims to shorten the path as much as possible. Since the planned path is coded by grid coordinates, the path length is the sum of the path lengths calculated from the ordered coordinate points, as shown in Eq. (2).

$$f_{a1} = \sum_{i=1}^{n-1} d\left(P_i, P_{i+1}\right)$$

$$d\left(P_i, P_{i+1}\right) = \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2 + (Z_{i+1} - Z_i)^2} \tag{2}$$

where $P_{\mathrm{i}}$ and $P_{\mathrm{i}} + 1$ are point $P_{\mathrm{i}}$ and point $P_{\mathrm{i}} + 1$, $d(P_{\mathrm{i}}, P_{\mathrm{i}} + 1)$ is the distance between the $i$-th and $i + 1$ points, $X_i$, $Y_i$, and $Z_i$ represent the three-dimensional coordinates of point i, $X_{i+1}$, $Y_{i+1}$, and $Z_{i+1}$ represent the three-dimensional coordinates of the $i + 1$ point, and $f_1$ is the value of the fitness function of the track length.

#### 2.3.2. Airway smoothness

The fitness function also aims to calculate the smoothness of the trajectory on the decision path. The smoothness can be calculated by two adjacent coordinate points, as shown in Eq. (3).

$$\alpha_i = atan\left[\frac{(X_{i-1} - X_i)\,(X_{i+1} - X_i) + (Y_{i-1} - Y_i)\,(Y_{i+1} - Y_i) + (Z_{i-1} - Z_i)\,(Z_{i+1} - Z_i)}{d\left(P_{i-1}, P_i\right) * d\left(P_{i+1}, P_i\right)}\right]$$

$$f_{a2} = \sum_{i=2}^{n-1} (\pi - \alpha_i) \tag{3}$$

where $\alpha_i$ is the route smoothness of two adjacent coordinate points, atan is the tangent function, $X_{\mathrm{i}}$, $Y_{\mathrm{i}}$, and $Z_{\mathrm{i}}$ represent the three-dimensional coordinates of point $i$, $X_{i+1}$, $Y_{i+1}$, and $Z_{i+1}$ represent the three-dimensional coordinates of point $i+1$, $d(P_{\text{i-1}}, P_{\text{i}})$ is the distance between the $i$-1 and $i$-th point, $\text{d}(P_{\text{i}}, P_{\text{i+1}})$ is the distance between the $i$-th and $i+1$ point, and $f_2$ is the smoothness fitness function value.

### 2.3.3. Airway concealment

The fitness function is designed to calculate the degree of safety of the decision path. The concealment is related to the flight height of the route and the distance between the route and the obstacle. The safety distance is specified in the coordinate system. When the safety distance is exceeded, the degree of concealment decreases; when the distance is lower than the safety distance, the degree of concealment increases. At the same time, it is proved that the movement has not reached the optimal moving distance, and the degree of concealment will be reduced in the next movement and increases the moving distance. The calculation is shown in Eq. (4).

$$S_i = \left(3 * \left(h_{\max} + d_{safe}\right)\right) * \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2}$$

$$S_j = \left(Z_{i+1} - d_{safe}\right) * \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2} \tag{4}$$

$$f_{a3} = \begin{cases} \sum_{i=2}^{n-1} S_i, d\left(P_i, P_{i+1}\right) < d_{safe} \\ \sum_{i=2}^{n-1} S_j, d\left(P_i, P_{i+1}\right) > d_{safe} \end{cases}$$

where $h_{max}$ represents the height of the highest point in the environment, $d_{safe}$ is a preset constant representing the safe distance, $X_i$, $Y_i$, and $Z_i$ represent the three-dimensional coordinates of point $i$, $X_{i+1}$, $Y_{i+1}$, and $Z_{i+1}$ represent the three-dimensional coordinates of point $i+1$, $d(P_{\text{i-1}}, P_{\text{i}})$ is the distance between the $i-1$ and $i$-th point, $d(P_{\text{i}}, P_{\text{i+1}})$ is the distance between the $i$-th and $i+1$ points, and $S_i$, $S_{j,}$ and $f_3$ are all the concealment fitness function values under the corresponding conditions

### 2.3.4. Airway threat

The fitness function is designed to calculate the threat degree between the decision path and the obstacle. The threat degree is related to the obstacle area that the path passes and the distance between the path and the obstacle center. To simplify the calculation, the obstacle is represented by a sphere. The calculation is shown in Eq. (5).

$$W_t = \sqrt{(X_i - X_t)^2 + (Y_i - Y_t)^2 + (Z_i - Z_t)^2}$$

$$f_{t_i} = \begin{cases} \sum_{t=1}^{k} W_t, & W_t < R_t \\ 0, & W_t > R_t \end{cases} \tag{5}$$

$$f_{a4} = \sum_{t=1}^{n} f_{t_i}, \qquad t_i = 1, 2 \ldots, n$$

$W_t$ represents the distance from any point $(X_{\text{i}}, Y_{\text{i}}, Z_{\text{i}})$ on the path to the center point of the obstacle $t(X_{\text{t}}, Y_{\text{t}}, Z_{\text{t}})$, $R_t$ represents the radius of the obstacle, and $f_t$ represents the threat of the t-th obstacle to the path. $k$ indicates that the current is the $k$-th obstacle, $n$ is the number of obstacles, and $f_4$ is the total threat fitness function value of all $n$ obstacles to the path.

Inspired by ref. [20], a weight is set for the fitness function of each cost. The weight represents the importance of the cost. The weight of each cost can be adjusted according to the actual situation to achieve a better adaptation to the environment. It is found that the overall weight size does not exceed 1 is the most appropriate, that is, $\Sigma\lambda_i = 1$. The weight of each cost can be adjusted according to the actual situation to achieve a better adaptation to the environment. According to the target weight method, under the condition of $\Sigma\lambda_i = 1$, the value of $\lambda_i$ can be fixed or random, and we find the appropriate

https://doi.org/10.1017/S0263574722001680 Published online by Cambridge University Press

value through continuous experiments. We set the sum of all λ as 1, because few people have mentioned influencing factors other than speed and height, so at the beginning of the test, we can only randomly generate different λ weights for four λ, in order to find a more suitable λ value, and then based on this value to carry out the current test. We ran about a dozen tests and picked the one with the lowest number. Therefore, the calculation for the fitness evaluation function of the path is shown in Eq. (6).

$$\begin{gathered} f=\lambda_{1}f_{a1}+\lambda_{2}f_{a2}+\lambda_{3}f_{a3}+\lambda_{4}f_{a4} \\ \lambda_{1}+\lambda_{2}+\lambda_{3}+\lambda_{4}=1 \end{gathered} \tag{6}$$

$f$ is the weighted sum of the fitness of the path, and $\lambda_{\mathrm{i}}$ are the weighting parameters of the four fitness values, where $i=1, 2, 3, 4$.

## 3. IB-ABC algorithm

### 3.1. Single way point employed bee strategy considering feedback information

Employed bee strategy refers to sharing information with a certain flight path code array and randomly selected flight paths through the process of crossover and mutation in order to update the flight path code array represented by the nectar [18]. Every time the employed bee strategy is used for optimization, it calls the employed bee to perform a search. The optimization result is the nectar source-optimized UAV flight path corresponding to the employed bee, referred to as the employed bee optimized path.

In this paper, the UAV path optimization process is carried out by considering the internal feedback information *Tr* (invalid search times) of the iteration. The path information sharing and optimization process is shown in Eq. (7).

$$\begin{gathered} X_i^{j*}=X_{\mathrm{i}}^{j}+\varphi(X_k^j-X_i^j)\mu(i), j\in[1,D]\,, k\in[1,SN] \\ \mu(i)=\frac{Tr(i)}{Tr(i)+Tr(k)}, Tr\in[1,D]\,, i\neq k \end{gathered} \tag{7}$$

$X_i^j$ is the $j$-th element of the $i$-th original UAV flight path, that is, the $j$-th coordinate point of the $i$-th UAV flight path, and $X_i^{j*}$ is the corresponding position of the employed bee optimization path optimized by the employed bee strategy. The path point coordinate *Tr(i)* is a constant, which represents the number of times that the path information is not updated after a round of path information optimization for the honey source $X_i$, and $\varphi$ is a random number between (−1,1). In this scheme, only a single waypoint $X_k^j$ is used each time to update the flight path of the UAV to ensure the local search ability. Every time the employed bee conducts a search but does not update the nectar source, *Tr* increases by 1.

After the employed bee strategy updates the path, the greedy selection strategy is executed to calculate the new nectar source amount, which is the fitness value of the $X^*{}_i$ employed bee optimized path obtained by Eq. (7). The fitness value of $X^*{}_i$ is compared with the fitness value of $X_i$. If the path fitness of $X^*{}_i$ is less than the fitness value of $X_i$, the employed bee moves to the new nectar source $X^*{}_i$. The performance is that the optimized path of the employed bee corresponding to $X^*{}_i$ replaces the flight path of the UAV corresponding to $X_i$, and vice versa, where the original UAV flight path $X_i$ is retained.

To sum up, Eq. (7) can guarantee the local search ability of the algorithm.

### 3.2. Two waypoint following bee strategy considering feedback information

The follower bee strategy refers to selecting a nectar source in a certain way and assigning a follower bee to it. Through the mutation process, a new flight path is searched around the UAV path corresponding to the nectar source [19]. Each time the follower bee strategy is used for optimization, it is called following the bee to perform a search. The optimization result is to follow the bee's corresponding nectar source to optimize the flight path of the drone, which is referred to as the follower bee optimization path.

In this paper, the method of directly assigning SN follower bees to the nectar source to search in parallel according to the degree of fitness is used, which is further refined as follows.

The calculation of the fitness value ratio $P$ for each UAV path is shown in Eq. (8).

$$P_{\mathrm{i}} = \frac{fitness(i)}{\sum_{j=1}^{SN} fitness(j)}$$

$$fitness(i) = \begin{cases} \dfrac{1}{1+f(i)} & f(i) \geq 0 \\ 1 + abs(f(i)) & f(i) \leq 0 \end{cases} \tag{8}$$

$f(i)$ represents the fitness function value of the optimized path of the $i$-th hired bee.

By calculating the fitness value ratio between the original flight path and the flight path of the employed bee retained after each employed bee strategy, all the nectar sources are sorted in descending order of the ratio, and a nectar source is randomly selected in order to allocate it following the bee until each follower bee is assigned a nectar source. In this way, employed bees with a larger fitness ratio can be assigned to follower bees earlier. The flight path of the UAV with better adaptability can prioritize the follower bee strategy to find the optimal path around.

In this article, the employed bee strategy has searched and optimized around the UAV path in Section 3.1. In the follower bee strategy, a larger search range than the employed bee strategy is used to optimize the path in a larger local range. The search method is shown in Eq. (9).

$$X_{i\ 0}^{j*} = X_m^j + \varphi(X_k^j - X_{i0}^j)\mu(i)$$

$$\mu(i) = \frac{Tr(i)}{Tr(i) + Tr(k)} \tag{9}$$

$$j \in [1, D], m, k \in [1, SN], Tr \in [1, D], i \neq k$$

$X_{i0}^j$ is the $j$-th element of the optimized path of the $i$-th employed bee, $X_{i\ 0}^{j*}$ is the path point corresponding to the position of the optimized path of the follower bee after executing the follower bee strategy, and $\varphi$ is a random number between (−1,1). In this scheme, in order to enhance the ability to search for an optimized flight path in the global scope, two elements $X_m^j$ and $X_k^j$ are used to update $X_{i0}^j$.

After the follower bee strategy, the greedy selection strategy is performed again. If the fitness value of the follower bee optimized path is less than the corresponding employed bee optimized path generated by the employed bee strategy, the corresponding employed bee flight path or the original flight path is deleted, and the follower bee optimized path is kept, which is regarded as moving the corresponding employed bee to the corresponding nectar source following the optimized path of the bee. Otherwise, the optimized path of the employed bee is retained.

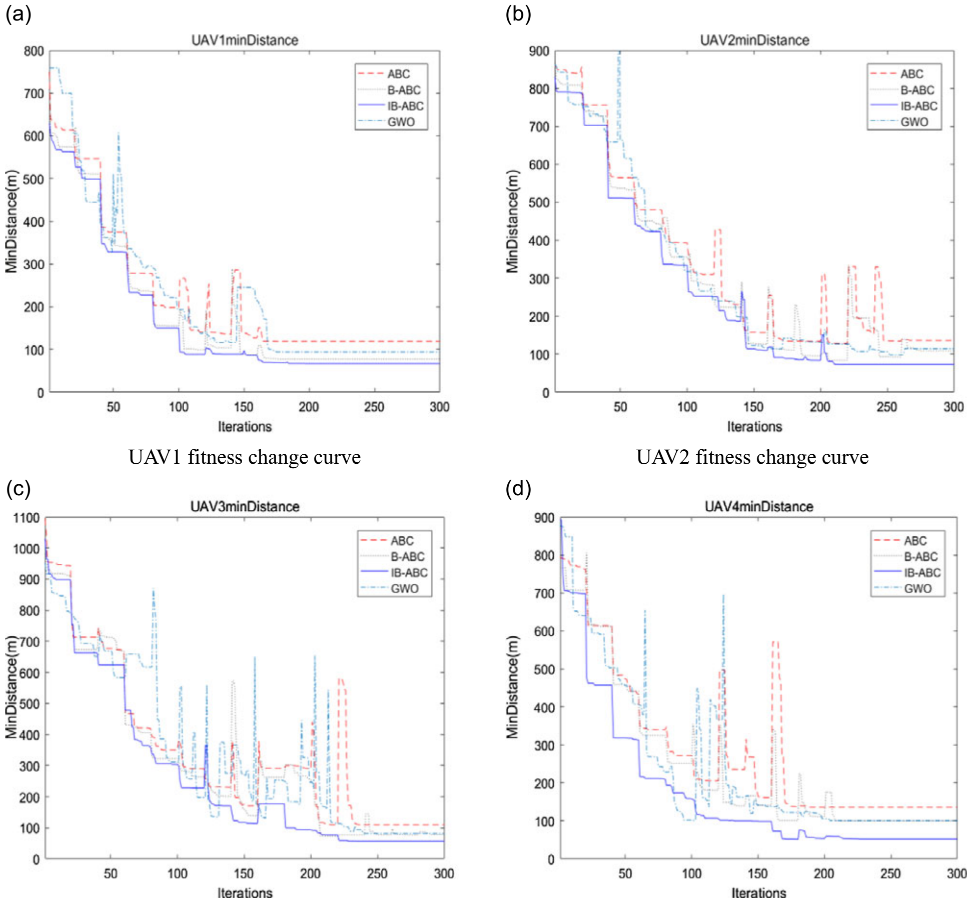
To sum up, Eq. (9) can enhance the global search ability of the algorithm and make the algorithm jump out of the local minimum.

### 3.3. Scout strategy with restrictions

The scout bee strategy refers to that in the iterative process. If a nectar source has gone through the employed bee strategy and the follower bee strategy and has not been updated, the drone path is deleted and a different UAV flight path is randomly generated globally, which then enters the iterative process again for optimization [20]. Each time the scout bee strategy is used to optimize the path, the scout bee replaces the employed bee to search. The optimization result is the nectar-optimized UAV flight path corresponding to the scout bee, referred to as the optimized path of the scout bee. Since this strategy re-generates a different UAV flight path globally, it can effectively avoid the algorithm falling into the local optimal value and increase the global search ability of the algorithm.

***Table I.*** *Simulation experiment parameters.*

| Experimental parameters | Value |
|---|---|
| Simulation environment size | 500 m × 500 m × 1000 m |
| UAV speed range | [0,30] m/s |
| Number of nectar sources | 20 |
| Scaling factor $\varphi$ | 0.5 |
| Limit altitude during flight | 40 m |



UAV1 fitness change curve

UAV2 fitness change curve

UAV3 fitness change curve

UAV4 fitness change curve

***Figure 4.*** *The adaptation curve of the drone group.*

In the iterative process, a Tr value is set for the flight path of the drone corresponding to each nectar source (Tr is defined in Eq. (7)), which is initially 0. After an employed bee strategy optimization, if the original drone flight path is not replaced by the employed bee optimization path or the follower bee strategy optimization, the employed bee optimization path and the original drone flight path are not replaced by the follower bee optimization path. Then the *Tr* value corresponding to the original UAV path is increased by 1. On the contrary, when the original UAV flight path corresponding to the $i$-th nectar source is replaced by the optimized path of the employed bee, the optimized path of the follower bee, or the optimized path of the employed bee is replaced by the optimized path of the follower bee, the value of corresponding UAV flight path $Tr(i)$ is reset to 1.

https://doi.org/10.1017/S0263574722001680 Published online by Cambridge University Press

***Table II.*** *Comparison of route data.*

| | ABC | | | B-ABC | | | IB-ABC | | |
|---|---|---|---|---|---|---|---|---|---|
| UAV | Path length | Planning time | Average speed | Path length | Planning time | Average speed | Path length | Planning time | Average speed |
| 1 | 272.164 | 15.952 | 17.0613 | 262.641 | 14.925 | 17.5971 | 261.670 | 14.249 | 18.3642 |
| 2 | 349.579 | 15.952 | 21.9142 | 349.121 | 14.925 | 23.3913 | 349.734 | 14.925 | 24.5445 |
| 3 | 423.873 | 15.952 | 26.5716 | 429.754 | 14.925 | 28.7938 | 327.469 | 14.925 | 22.9818 |
| 4 | 327.434 | 15.952 | 20.5260 | 307.144 | 14.925 | 20.5788 | 301.035 | 14.925 | 21.1268 |

After each iteration, any $Tr(i)$ that exceeds the threshold Th (a constant set in advance, used to measure the size of $Tr$) will be reset to $L$, instead of directly implementing the scout bee strategy. Before the next iteration, compare the average value $\overline{T}$ of $Tr$ with the size of $\gamma^{*}D$ ($\gamma\in(0,1)$, which is a given parameter to determine the degree of invalid search and $D$ is the path point in each path. If $\gamma^{*}D$ is small, it means that the effective operation efficiency of the current iterative system is too low, and the overall update needs to be performed. At this time, the scout bee strategy is implemented to delete any number of $\gamma^{*}SN$ drone flight paths, including the original drone flight path that has not been optimized, the optimized path of employed bees and the optimized path of the follower bees, initialize the same number of optimized paths of scout bees, reset the corresponding $Tr$ of $\gamma^{*}SN$ nectar source paths to 1, and enter the next iteration. If $\gamma^{*}SN$ is large, go directly to the next iteration and retain all current drone flight paths, including the initial drone flight path that has not been optimized, the employed bee optimization path, and the follower bee optimization path. Do not generate an optimized path for scout bees. The calculation of $\overline{T}$ value is shown in Eq. (10).

$$\overline{T}= (1/SN)\sum_{i=1}^{SN} Tr\,(i) \tag{10}$$

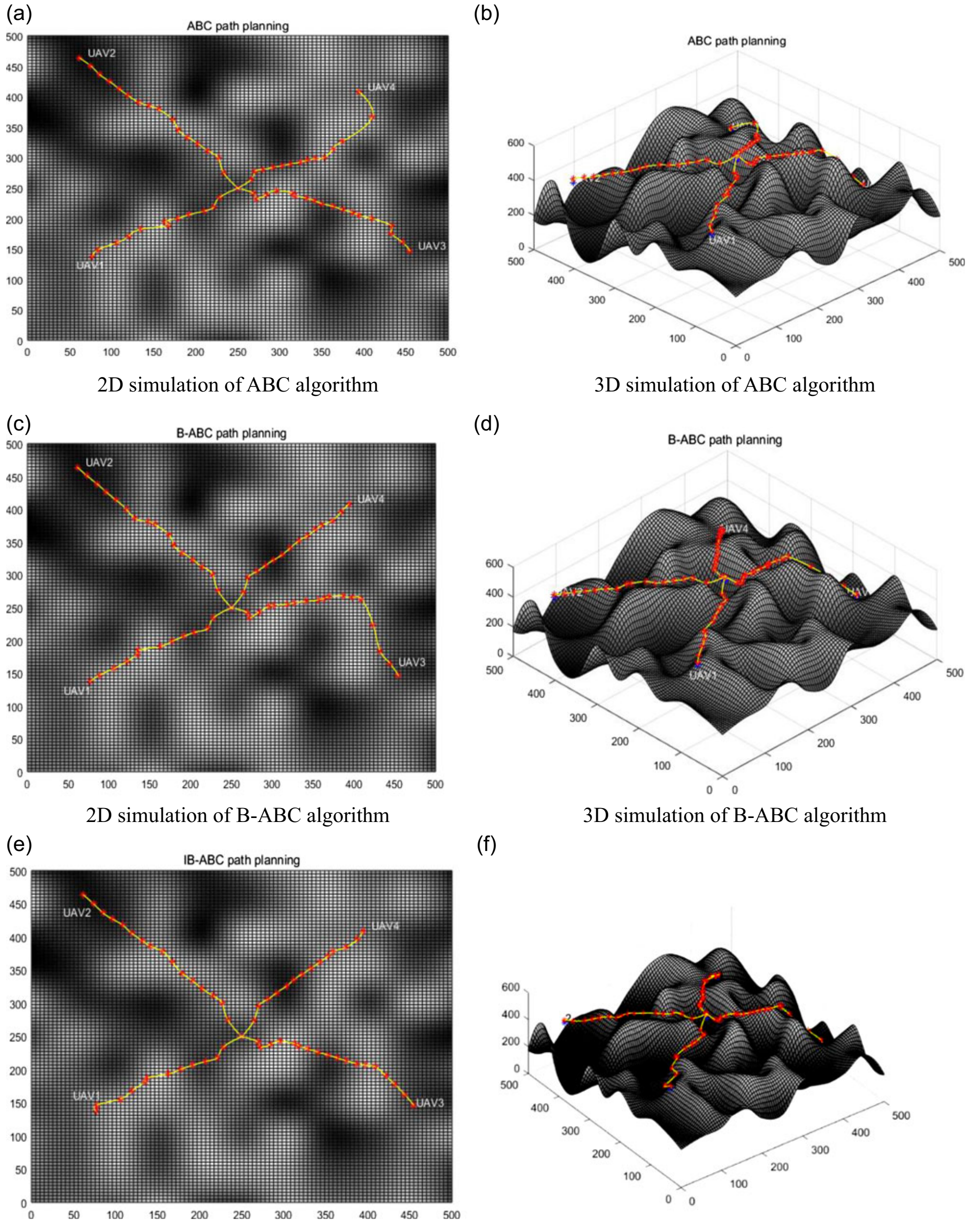
$\overline{T}$ represents the average $Tr$ of $SN$ honey source paths and $Tr(i)$ represents the number of times that the $i$-th honey source has gone through a round of path optimization without updating the path information so far.

## 4. Simulation experiment results and analysis

### *4.1. Multiple drones depart and arrive at the same time*

In order to verify the performance of the IB-ABC algorithm in the UAV path planning problem, the simulation experiment parameters are set as shown in Table I. The limit height is the maximum distance that the UAV can fly. The UAV cannot exceed this height during the flight process. Combined with practical problems, this height is set in the simulation experiment. The four experimental objects are set to UAV1-UAV4, the starting point is set to (75, 135, 400), (60, 450, 350), (450, 145, 250), (400, 400, 350), the end point is set to (250, 250, 550), the IB-ABC algorithm is in the experimental group, and the ABC algorithm, the balanced ABC (B-ABC), and gray wolf algorithm are the control group. Among them, the number of populations in the gray wolf algorithm is 20, and the number of iterations is the same as the ABC-related algorithm. Other related parameters are the same as the ABC-related algorithm. The adaptation curve of each UAV is shown in Fig. 4, and the simulation result is shown in Fig. 6. The average value of 10 random experiments is used to obtain a comparison table of algorithm route data, as shown in Table II.

From the path optimization convergence curve (Fig. 4(a)–(d)), it can be seen that when the UAV group is required to reach the target location at the same time, the ABC algorithm is due to the imbalance problem of its global and local search capabilities, at the end of each iteration cycle, the detection bee strategy will cause the optimization result to mutate in a larger random direction, causing the planned



2D simulation of ABC algorithm

3D simulation of ABC algorithm

2D simulation of B-ABC algorithm

3D simulation of B-ABC algorithm

2D simulation of IB-ABC algorithm

3D simulation of IB-ABC algorithm

***Figure 5.** Algorithm simulation comparison chart.*

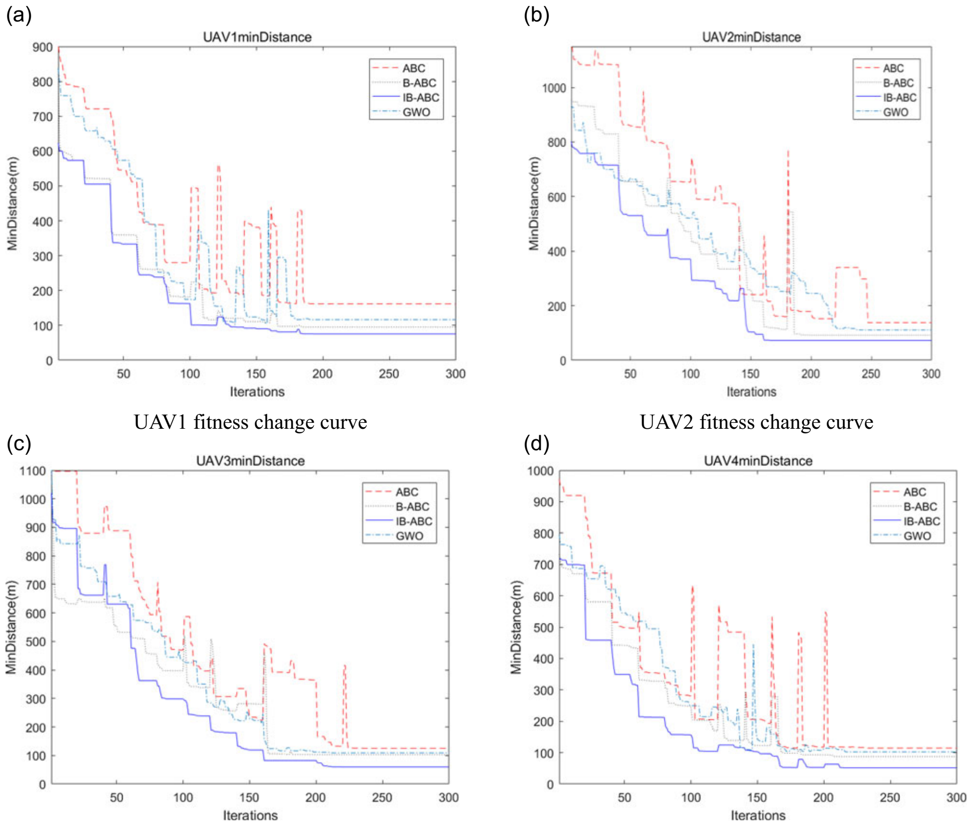
path to iterate in the direction of deterioration, which is specifically manifested as a sharp increase in the path length, and the sharp increase is greater. The B-ABC algorithm reduces the surge and probability of the planned path after the end of the iteration cycle by using a balanced strategy. The IB-ABC algorithm further reduces the randomness of variation and speeds up the convergence speed by considering the internal information of the iteration. Its convergence speed and path length are also better than the gray wolf algorithm.

Based on the two-dimensional and three-dimensional flight trajectories in Fig. 5(a)–(f), the IB-ABC algorithm has improvement in obstacle avoidance, path smoothness, and path length when compared to the ABC and B-ABC algorithms.

https://doi.org/10.1017/S0263574722001680 Published online by Cambridge University Press

*Table III. Simulation experiment parameters.*

| Experimental parameters | Value |
|---|---|
| Simulation environment size | 500 m × 500 m × 1000 m |
| UAV speed range | [3,30] m/s |
| Iteration cycle | 20 |
| Number of nectar sources | 20 |
| Scaling factor | 0.5 |
| Limit altitude during flight | 40 m |
| Flight time interval | 9 s |



UAV1 fitness change curve

UAV2 fitness change curve

UAV3 fitness change curve

UAV4 fitness change curve

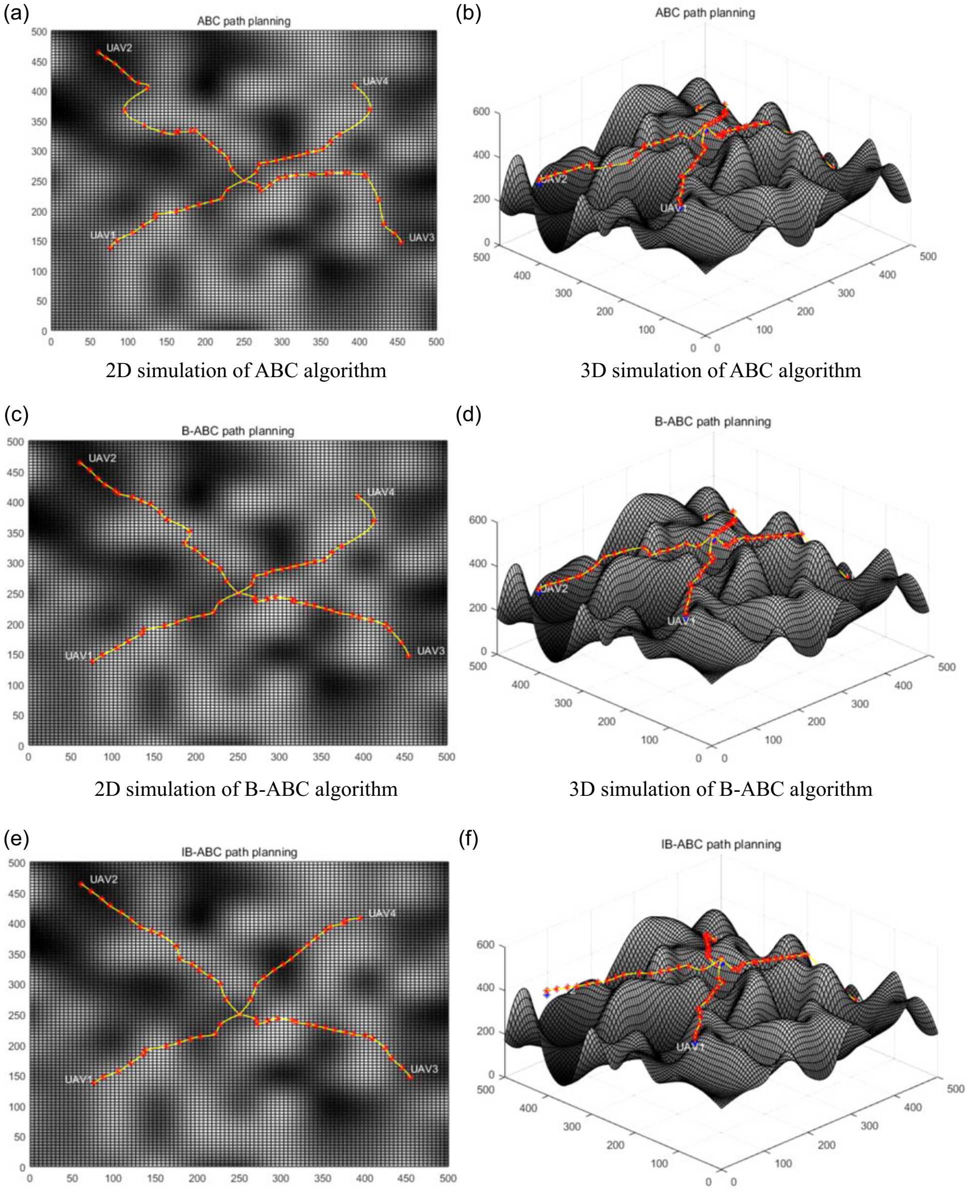
***Figure 6.*** *Adaptation curve of UAV group.*

Table II shows that the B-ABC and IB-ABC algorithms shorten the path length by 3% and 10%, respectively, when compared with the ABC algorithm. They also reduce the flight time by 7% and 11%, respectively. Since all UAVs depart and arrive at the same time, each UAV has the same planned time. The speed in the table is the average speed, which is calculated by dividing the total distance by the time.

### *4.2. Multiple UAVs start at the same time and arrive in sequence*

In this group of experiments, the performance simulation experiment parameters that verify the IB-ABC algorithm in the UAV path planning problem are shown in Table III. The limit height is the maximum distance that the UAV can fly. The UAV cannot exceed this height during the flight process. Combined

https://doi.org/10.1017/S0263574722001680 Published online by Cambridge University Press



2D simulation of ABC algorithm

3D simulation of ABC algorithm

2D simulation of B-ABC algorithm

3D simulation of B-ABC algorithm

2D simulation of IB-ABC algorithm

3D simulation of IB-ABC algorithm

***Figure 7.*** *Algorithm simulation comparison chart.*

with practical problems, this height is set in the simulation experiment. The experimental subjects are set as four UAVs UAV1–UAV4, the IB-ABC algorithm as the experimental group, and the ABC algorithm and B-ABC algorithm as the control group. The adaptation curve of each UAV is shown in Fig. 6, and the simulation result is shown in Fig. 7. The average value of 10 random experiments is used to obtain a comparison table of algorithm route data, as shown in Table IV.

From the path optimization convergence curve (Fig. 6(a)–(d)), it can be seen that when each individual in the UAV swarm is required to arrive at the destination in a fixed order, the ABC algorithm's imbalance of its global and local search capabilities is more obvious, the degraded iteration phenomenon has increased significantly, and there is an optimization behavior based on the degraded path, which is

***Table IV.*** *Comparison of route data.*

| | ABC | | | B-ABC | | | IB-ABC | | |
|---|---|---|---|---|---|---|---|---|---|
| **UAV** | **Path length** | **Planning time** | **Average speed** | **Path length** | **Planning time** | **Average speed** | **Path length** | **Planning time** | **Average speed** |
| 1 | 262.65 | 8.7551 | 30 | 262.488 | 8.7496 | 30 | 261.115 | 8.7039 | 30 |
| 2 | 327.76 | 17.7496 | 18.4659 | 328.321 | 17.704 | 18.545 | 306.704 | 17.755 | 17.2739 |
| 3 | 452.35 | 26.7496 | 16.9106 | 432.470 | 26.704 | 16.195 | 350.520 | 26.755 | 13.1011 |
| 4 | 437.01 | 35.7496 | 12.2242 | 424.269 | 35.755 | 11.866 | 421.547 | 35.704 | 11.8068 |

specifically manifested as a large, sharp increase in the path length. The B-ABC algorithm reduces the surge and probability of the planned path after the end of the iteration cycle by using a balanced strategy. The IB-ABC algorithm further reduces the randomness of variation and increases the convergence speed by considering the internal information of the iteration. Its convergence speed and path length are also better than the gray wolf algorithm.

From the two-dimensional and three-dimensional flight trajectory figures (Fig. 7(a)–(f)), when each individual in the UAV swarm is required to arrive at the destination in a fixed order, the IB-ABC algorithm has significant improvements in planning paths, path length, and optimization degree when compared to the ABC and B-ABC algorithms.

It can be seen from Table IV that, because the B-ABC algorithm concentrates on improving the balance ability of local and global search, there is little improvement in optimizing path parameters, while the IB-ABC algorithm considers the internal information of the iteration. Compared with the ABC algorithm, the IB-ABC algorithm improves the path length by 7.4%. It should be noted that no matter which algorithm is under the control, the speed of UAV1 is always maintained at 30 m/s, which is normal. The setting requires UAV1 to arrive first, and without a given arrival time, UAV1 flies at the speed limit. The speed in the table is the average speed, which is calculated by dividing the total distance by the time.

## 5. Conclusions

In regard to the problem of UAV path planning, this paper proposes the IB-ABC algorithm. By using the balanced strategy and the iterative internal information feedback to optimize the path, the local search ability and the global search ability are coordinated, the probability of large random variation in the iteration process is reduced, and the optimization process is ensured to proceed in a better direction. The experiments of this paper are carried out by simulation experiments, and there is no practical flight experiment. The simulation experiment results prove that the IB-ABC algorithm improves the optimization efficiency, shortens the planning path, increases the iteration speed, and can effectively realize the path planning in the complex mountain environment.

**Author contributions.** Conceptualization, Li Tan; formal analysis, Jiaqi Shi; funding acquisition, Li Tan and Yu Zhang; methodology, Jiaqi Shi and Hongtao Zhang; supervision, Li Tan and Jing Gao; validation, Li Tan; Writing – original draft, Jiaqi Shi and Tianying Xu; writing – review and editing, Li Tan and Xiaofeng Lian.

**Financial support.** This research was funded by the National Natural Science Foundation of China (61702020) and Fundamental Research Funds for Beijing Universities of Civil Engineering and Architecture (X20069).

**Conflicts of interest.** The authors declare no conflicts of interest exist.

**Ethical approval.** Not applicable.

## References

[1] P. Fraga-Lamas, L. Ramos, V. Mondéjar-Guerra and T. M. Fernández-Caramés, “A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance,” *Remote Sens.* **11**(18), 2144 (2019).

[2] Y. Zhou, P. Wang, X. Li and K. Zhang, “Cooperative path planning of multi-UAV based on multi-objective optimization algorithm,” *Syst. Eng. Electron*. **39**(4), 782–787 (2017).

[3] C. Ntakolia and D. K. Iakovidis, “A swarm intelligence graph-based pathfinding algorithm (SIGPA) for multi-objective route planning,” *Comput. Oper. Res*. **133**, 105358 (2021).

[4] S. Konatowski and P. Pawłowski, “Ant Colony Optimization Algorithm for UAV Path Planning,” **In:** *14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering* (IEEE, Ukraine, 2018) pp. 177–182.

[5] U. Orozco-Rosas, O. Montiel and R. Sepúlveda, “Mobile robot path planning using membrane evolutionary artificial potential field,” *Appl. Soft Comput. J*. **77**, 236–251 (2019).

[6] M. Xu, W. Wang, H. Wang, S. Xiao and Z. Huang, “Multipopulation artificial bee colony algorithm based on a modified probability selection model,” *Concurr. Computat*. **33**(13), e6216 (2021).

[7] H. Chiang and A. K. Sangaiah, “A novel artificial bee colony optimization algorithm with SVM for bio-inspired software-defined networking,” *Int. J. Parallel Prog.* **48**(2), 310–328 (2020).

[8] S. F. Hussain, A. Pervez and M. Hussain, “Co-clustering optimization using artificial bee colony (ABC) algorithm,” *Appl. Soft Comput.* **97**, 97 (2020).

[9] B. Alatas, “Chaotic bee colony algorithms for global numerical optimization,” *Exp. Syst. Appl.* **37**(8), 5682–5687 (2010).

[10] U. Deniz and A. Ali, “A Study on the Performance of the Hybrid Optimization Method Based on Artificial Bee Colony and Differential Evolution Algorithms,” **In:** *International Artificial Intelligence and Data Processing Symposium* (IEEE, Turkey, 2017).

[11] Z. Li, T. Liu, H. Li and Z. Sun, “Orthodontic Path Planning Method Based on Optimized Artificial Bee Colony Algorithm,” **In:** *2020 5th International Conference on Intelligent Computing and Signal Processing, ICSP 2020* (IOPscience, China, 2020).

[12] R. Privarova,"Operations Research as a Tool for Optimization of Transport Routes in Crisis Planning,” **In:** *Proceedings of the 20th International Scientific Conference Transport, MEANS 2016*(2016), pp. 798–803.

[13] Z. Xian, R. Yaofeng and W. Runpeng, “Research on optimal search path programming in continuous tine and space based on adaptive genetic algorithm,” *Acta Armamentarii* **36**, 225(12), 2386–2395 (2015).

[14] M. Konar, A. Turkmen and T. Oktay, “Improvement of the thrust-torque ratio of an unmanned helicopter by using the ABC algorithm,” *Aircr. Eng. Aerosp. Technol.* **92**(8), 1133–1139 (2020).

[15] S. Aslan, “A comparative study between artificial bee colony (ABC) algorithm and its variants on big data optimization,” *Memet. Comput.* **12**(2), 129–150 (2020).

[16] M. Konar, “Simultaneous determination of maximum acceleration and endurance of morphing UAV with ABC algorithm-based model,” *Aircr. Eng. Aerosp. Technol.* **92**(4), 579–586 (2020).

[17] R. Lu, H. Hu, M. Xi, H. Gao and C. Pun, “An improved artificial bee colony algorithm with fast strategy and its application,” *Comput. Electr. Eng*. **78**, 79–88 (2019).

[18] M. Sood and V. K. Panchal, “Meta-heuristic techniques for path planning: Recent trends and advancements,” *Int. J. Intell. Syst. Technol. Appl*. **19**(1), 36–77 (2020).

[19] A. Nayyar, N. N. guyen, R. Kumari and S. Kumar, “Robot Path Planning Using Modified Artificial Bee Colony Algorithm,” **In:** *Proceedings of the 7th International Conference on FICTA 2018*, Singapore, vol. **1014** (2018) pp. 25–36.

[20] Z. Li, Z. Zhang, H. Liu and L. Yang, “A new path planning method based on concave polygon convex decomposition and artificial bee colony algorithm,” *Int. J. Adv. Robot. Syst.* **16**(6), 172988141989478 (2020).

[21] F. Xu, H. Li, P. c, H. Hu, Y. Li, Y. Song and H. Gao, “A new global best guided artificial bee colony algorithm with application in robot path planning,” *Appl. Soft Comput.* **88**, 106037 (2020).

[22] C. Wu, X.Huang, Y. Luo and S. Leng, “An Improved Fast Convergent Artificial Bee Colony Algorithm for Unmanned Aerial Vehicle Path Planning in Battlefield Environment,” **In:** *2020 IEEE 16th International Conference on Control & Automation (ICCA)* (IEEE, Singapore, 2020).

**Cite this article:** L. Tan, J. Shi, J. Gao, H. Wang, H. Zhang and Y. Zhang (2023). “Multi-UAV path planning based on IB-ABC with restricted planned arrival sequence”, Robotica **41**, 1244–1257. https://doi.org/10.1017/S0263574722001680