Three-Dimensional UAV Path Planning in Urban Environments Based on an Improved Parrot Optimization Algorithm

Hongtao Ma[1*] Zihang Zhou[1*] Yisheng Gao[2*]
1. Beijing International Studies University 2.Shangdong University of Technology

Abstract: With the rapid development of unmanned aerial vehicle (UAV) technology, its applications in military and civilian fields are becoming increasingly widespread. However, realizing autonomous navigation of UAVs in complex environments still faces numerous challenges, especially the three-dimensional path planning problem. This paper proposes an improved parrot optimization algorithm (IPO) by introducing SPM chaotic mapping, adaptive switching factor, and hybrid Cauchy and Gaussian mutation strategies to enhance the algorithm's global search capability and convergence speed. The improved IPO is combined with the MATLAB simulation platform to construct a complete three-dimensional path-planning solution framework. Extensive simulation experiments demonstrate that, compared with the standard parrot optimization and other optimization algorithms, this algorithm significantly improves optimization accuracy, convergence speed, and path smoothness, showing good potential for engineering applications. The experimental results indicate that under the complex three-dimensional environment modeling of the Beijing International Studies University (BISU) campus, the improved IPO algorithm can quickly and accurately find the optimal collision-free path with minimal flight cost, which demonstrates excellent prospects for engineering applications.

Keywords: UAV three-dimensional path planning, improved parrot optimization algorithm, SPM chaotic mapping, adaptive switching factor, hybrid Cauchy and Gaussian mutation

1. Introduction

Introduction With the rapid development of unmanned aerial vehicle (UAV) technology, its applications in fields such as military reconnaissance, disaster relief, and logistics have become increasingly widespread [1]. In complex and dynamic environments, the autonomous flight path of UAVs has become a hot topic and challenge in current research. The core challenge lies in how to accurately plan a three-dimensional trajectory that is both collision-free and minimizes flight cost, which directly relates to the quality of the UAV's autonomous decision-making capability[2].

Traditional path planning algorithms, such as Dijkstra, A* search, and Rapidly-exploring Random Tree (RRT), often employ heuristic search or random sampling to explore feasible solutions in the state space. Karaman et al. [3] were among the first to introduce the RRT algorithm into UAV path planning, achieving higher computational efficiency than traditional methods, but the path quality needs improvement. Roberge et al. [4] proposed an improved genetic algorithm (GA) that achieved good results in UAV trajectory optimization by carefully designing crossover and mutation operators. However, these algorithms often struggle to find satisfactory solutions within a limited time when dealing with large-scale, high-dimensional problems and suffer from high computational complexity and blind search [5]. Therefore, Yan Md et al. [6] introduced an enhanced particle swarm optimization algorithm (GA-PSO) that introduces partial matching crossover and secondary swap mutation to solve the problems of random task allocation and two-dimensional path planning in multi-UAV systems. Cabreira T M et al. [7] explored the advantages of different flight modes and cell decomposition methods under different scenario shapes, discussed the coverage performance of UAVs under different information conditions, and pointed out the challenges of improving the efficiency and robustness of multi-UAV collaboration in future research.

In recent years, swarm intelligence optimization algorithms have received increasing attention in UAV path planning due to their good global search capability and robustness. Zhao et al. [8] proposed a PSO-based real-time obstacle avoidance technique for UAVs, which performs well in specific situations but quickly falls into local optima in complex environments. To address this, Fu et al. [9] designed a hybrid quantum-behaved PSO algorithm that significantly improves

[1] Hongtao Ma & Zihang Zhou, Beijing International Studies University, School of Economics. Class 2020.
[2] Yisheng Gao,Shandong University of Technology,School of Management. Class 2022
* Equal contribution, all as first author, Hongtao Ma is corresponding author, email: 2020221242@stu.bisu.edu.cn

solution quality and convergence speed, but the precise adjustment of algorithm parameters remains challenging. Yang et al. [10] introduced the bat algorithm (BA) into UAV three-dimensional path planning, leveraging its powerful global search capability but lacking path smoothness. Wang et al. [11] improved the firefly algorithm (FA) by introducing adaptive step size and Lévy flight, significantly optimizing the search performance in UAV trajectory planning, but it has not been fully validated in complex real-world environments. Huang et al. [12] proposed a hybrid ABC/ACO algorithm that demonstrates superiority over single intelligent algorithms in the field of UAV path planning, especially in convergence speed and solution smoothness, but its computational complexity is relatively high. Therefore, Alejo D et al. [13] proposed an improved three-dimensional optimal reciprocal collision avoidance algorithm (3D-ORCA) for real-time collision avoidance of multi-UAV systems in dynamic and static obstacle environments. Compared to the traditional ORCA algorithm, this method enhances the handling of static obstacles and considers the dynamic constraints of UAVs. By integrating it into the ROS framework and conducting various simulation tests, the effectiveness of the algorithm in complex environments is verified. Shao Z, Yan F, Zhou Z et al. [14] proposed a UAV swarm path planning method based on a distributed cooperative particle swarm optimization algorithm (DCPSO) to achieve simultaneous arrival and formation aggregation of UAVs at designated locations. Zhu et al. [15] designed a distributed improved ant lion optimization algorithm (MALO) for multi-UAV collaborative tasks, efficiently realizing task allocation and conflict avoidance, but its application scenarios still have certain limitations. Iacono M et al. [16] proposed an autonomous UAV path following and obstacle avoidance method based on a depth camera. This method uses a depth camera to generate an environment map and adjusts the path in real time when the UAV detects obstacles, ensuring the effectiveness of obstacle avoidance and path following. Experimental results show that the method can effectively navigate in complex indoor environments with high computational efficiency and scalability. It is suitable for various UAV autonomous navigation applications. Qu et al. [17] successfully applied an improved whale optimization algorithm (WOA) to complex three-dimensional space trajectory planning for UAVs, significantly improving convergence speed and search accuracy compared to traditional intelligent algorithms. Zhang et al. [18] combined grey wolf optimization (GWO) with differential evolution (DE) to form a hybrid meta-heuristic algorithm, which achieved better optimization effects than single intelligent algorithms in online UAV trajectory optimization. Phung M D et al. [19] proposed an enhanced discrete particle swarm optimization (DPSO) algorithm for UAV vision-based surface inspection path planning. The paper formulates the path planning problem as an extended traveling salesman problem (TSP), enhancing the path while satisfying coverage and obstacle avoidance constraints. The algorithm is parallelized on GPUs by introducing deterministic initialization, random mutation, and edge exchange techniques to improve computational efficiency. Experimental results demonstrate significant path quality and computation speed improvements for large-scale structural examinations. The algorithm suits UAV inspection tasks in complex environments, such as bridges.

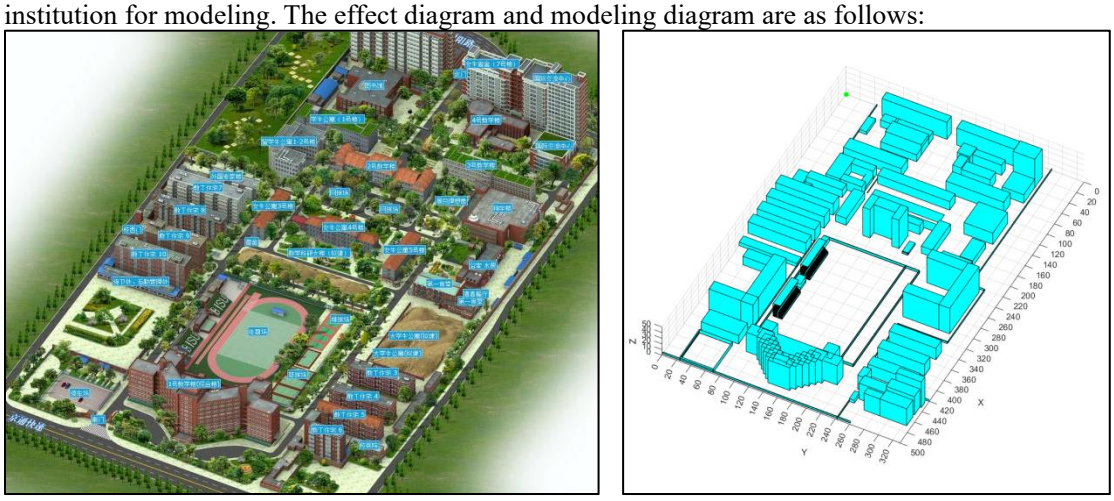
However, existing intelligent optimization algorithms still suffer from slow convergence speed, easily falling into local optima, and poor environmental adaptability, requiring further improvement. This paper proposes an improved parrot optimization algorithm (IPO) to address the above issues and applies it to UAV three-dimensional path planning tasks. In this paper, we introduce three strategies: SPM chaotic mapping, adaptive switching factor, and hybrid Cauchy/Gaussian mutation to address the shortcomings of the standard parrot optimization algorithm. Through a series of simulation experiments, we verify the effectiveness of the improved algorithm in path-planning tasks and compare its performance with several other mainstream intelligent optimization algorithms. Then, on this basis, we establish a mathematical model for UAV path planning, comprehensively considering multiple optimization objectives such as flight distance, energy consumption, and safety distance. Finally, we model the campus of the authors' communication unit, Beijing International Studies University (BISU), to construct a complex three-dimensional environment model and extract key waypoints as critical nodes for our path planning. Through experiments, we simulate its operation in a real environment.

2. Model Establishment

The three-dimensional path planning problem for UAVs in complex environments is essential. The UAV aims to start from a starting point, avoid obstacles, and reach a predetermined endpoint. This paper utilizes an improved parrot optimization (IPO) algorithm to explore an optimization and evaluation method for UAV path planning by establishing a three-dimensional environment model and designing a fitness function. This section will briefly introduce the design and implementation of the environment modeling and fitness function.

2.1 Map Selection

To further simulate the application of UAV path optimization in real life, this paper selects the communication unit of the corresponding author and the first author, Beijing International Studies University (BISU), for environment modeling. The reason for choosing this school for modeling is that it is a familiar living environment for the authors, and simulated flight has decisive practical significance for the authors' university. Moreover, as a university in Beijing, BISU has a relatively small area, making it easy to model. Therefore, we use the authors' institution for modeling. The effect diagram and modeling diagram are as follows:



A B

Figure 1: Architectural rendering of Beijing International Studies University[3] (A) and modeling result (B)

As shown in the figure, Figure A basically models all the buildings on the campus. In the model of Figure B, the values of length, width, and height of the corresponding buildings are presented in the modeling map on a one-to-one equal scale. It can reflect and simulate the spatial structure[4] of the school in reality.

2.2 Map Modeling and Design

First, we set the boundaries of the virtual map in the three-dimensional space. As shown in Figure 2, we refer to the proportions of length, width, and height in real life and extend from the origin$[0,0,0]$ to constrain the size of the map to$[500 \times 50 \times 50]$. The three values in the brackets represent the dimensional limits on the x, y, and z axes, respectively, corresponding to the general situation of 500 meters in length from north to south and 350 meters in width in real life.

For $[500 \times 50 \times 50]$, each length of 1 is called a physical unit.

As the map environment becomes larger, the experimental efficiency decreases. To balance efficiency and experimental feasibility, we define the boundary of the UAV flight as $[300 \times 300 \times 20]$, which is the range specified in Figure 2. In real life, the main living area of students is

[3] Due to certain changes at the university in recent years, the actual distribution of buildings may differ somewhat from the architectural rendering. Therefore, the model is primarily based on the actual situation, which may vary from the rendering but remains broadly similar.

[4] The dimensions (length, width, height) of the buildings are estimates made by the authors, which slightly differ from the actual values.

basically within the defined range, so the experimental results still have value. The map is divided into unit grids since obstacles should be represented in a discrete grid. Considering the final experimental efficiency, this paper sets the size of the map unit to $[2 \times 2 \times 2]$, that is:

$$unit = [2, 2, 2] \tag{1}$$

Dividing the set UAV flight boundary size $[300 \times 300 \times 20]$ by the map unit, we obtain the size of the new map after discretization, which is represented as a three-dimensional matrix of $[150 \times 150 \times 10]$. The simulated UAV will work within the defined flight range, translating one map unit, i.e., flying two physical units, on the x, y, and z axes each time. In real life, each displacement can be understood as moving two meters, which we consider a reasonable unit flight distance.

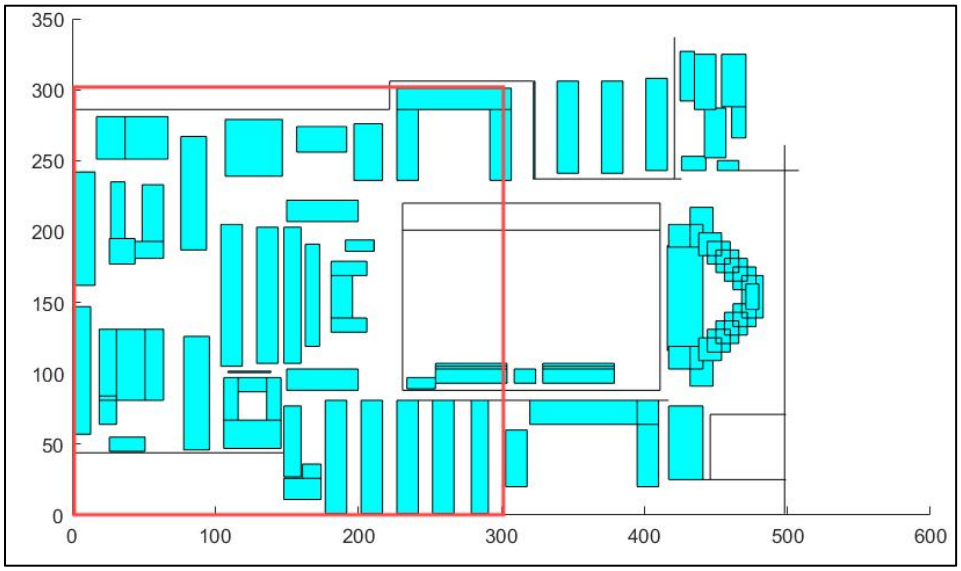


Figure 2: Floor plan of Beijing International Studies University (BISU) and the delineated experimental area

Then, we design the obstacles in the figure. To represent the initial state of the map (i.e., without obstacles), we define an initial all-zero matrix($Map_{Initial}$) with all values set to 0. Subsequently, based on our obstacle selection, we mark the corresponding grid units as 1 to indicate that obstacles occupy these positions. The obstacles in the map are defined in a parameterized way. The definition of each obstacle includes its position (X, Y, Z) and size (L, W, H). For example, the definition of an obstacle is as follows:

$$Obstacle(i) = [X_i, Y_i, Z_i, L_i, W_i, H_i] \tag{2}$$

For this matrix, each row represents an obstacle, where X_i, Y_i, Z_i represents the initial position of the obstacle in the map, denoted as $(x_{start}, y_{start}, z_{start})$ in the following text, and L_i, W_i, H_i represent its length, width, and height, respectively. The endpoint $(x_{\text{end}}, y_{\text{end}}, z_{\text{end}})$ of each obstacle is calculated accordingly. In the obstacle matrix, each input row adds a new obstacle to the map, and the parameters can be repeatedly adjusted to simulate the barriers, such as buildings, walls, and stands, in real-life scenarios.

Therefore, to map the obstacles to the discrete grid, we first need to divide the position and size of each $Obstacle$ by the size of the unit grid to obtain the starting position and size of the obstacle in the grid. Then, based on the starting coordinates $(x_{start}, y_{start}, z_{start})$ and ending coordinates $(x_{\text{end}}, y_{\text{end}}, z_{\text{end}})$ of the obstacle, we make the following judgment for all coordinates $\text{map}(x, y, z)$ in the map:

$$\text{map}(x, y, z) = \begin{cases} 1, & \text{if } x_{\text{start}} \le x \le x_{\text{end}},\ y_{\text{start}} \le y \le y_{\text{end}},\ z_{\text{start}} \le z \le z_{\text{end}} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

When the values of (x, y, z) are all between the starting and ending coordinates of the obstacle, it means that the position is within the range of the obstacle, so the value of

$map(x, y, z)$ is 1. Otherwise, the value of $map(x, y, z)$ is 0, indicating that the position is not occupied by an obstacle and is an empty unit grid that can be used for UAV flight.

2.3 Path Planning

Generally speaking, the goal of UAV three-dimensional path planning is to find the optimal path from the starting point S to the endpoint E in a complex environment through intelligent optimization algorithms while ensuring that the UAV avoids all obstacles along the path[5].

To better simulate the scenarios that UAVs may encounter in real life, in our simulation experiments, we add two additional experimental requirements on top of the original task of finding the optimal path from the starting point to the endpoint: passing through a specified mid-point and maintaining a specific flight attitude.

The choice of a mid-point considers the situation where UAV transportation may involve a traveling salesman problem, i.e., in real life, UAVs need to fly to multiple points to perform tasks in one go. Considering the experimental efficiency and practical requirements, this paper only designs one mid-point and divides the path planning into two segments, as follows:

First segment path planning (starting point to mid-point): In the first segment path planning, the starting point is used as the initial position, the mid-point as the temporary endpoint, and the best path is obtained through optimization using the IPO algorithm.

Second segment path planning (mid-point to endpoint): After completing the first segment path planning, the mid-point is used as the new starting point, the endpoint position as the final goal, and the second segment path planning is performed using the same IPO algorithm.

Through the above method, we can achieve path planning after the mid-point.

At the same time, the second requirement, setting a specified flight attitude, is considered for the safety of UAV flight in real life. In this experiment, we specify the safe flight attitude of the UAV to be 8. The specific derivation process is as follows:

Assuming a UAV without air resistance loses control and crashes, its parachute will open 1 second after sensing free fall. It will first undergo free fall, and its displacement is:

$$h_1 = \frac{1}{2} g t^2 = \frac{1}{2} \times 9.81 \times 1^2 = 4.905\text{m} \tag{4}$$

Where t = 1 second and g = 9.81 m/s². Within 1 second of sensing weightlessness, the UAV will fall 4.905 meters. After the parachute opens, the UAV will start to decelerate. But to avoid hitting people on the ground, assuming the height of a person is h_2 = 2 meters, the performance of the parachute can reduce the final velocity of the UAV to a safe speed that will not harm people within a certain falling distance after opening. Assuming this falling distance used for buffering is h_3 = 1 meter, then the total minimum safe altitude is:

$$h_{\min} = h_1 + h_2 + h_3 = 4.905 + 2 + 1 \approx 8\text{m} \tag{5}$$

Therefore, to simulate a safe environment, the UAV in this experiment will maintain a flight altitude of at least 8 meters to avoid hitting people before the parachute opens due to an excessively long free-fall time.

In summary, a complete UAV path planning process in this simulation experiment is as follows: After the UAV starts from the starting point, it rises vertically along the z-axis to reach the preset safe attitude. Then, it moves horizontally in the x-y plane at an attitude not lower than the safe attitude (8 in this paper). When the UAV horizontally reaches the x-y coordinates of the mid-point, it begins to descend vertically along the z-axis, then takes off again, repeats the above process, and reaches the z-coordinate of the endpoint.

2.4 Path Length Function Design

In the above text, each process of passing through the mid-point and finding the optimal path is split into two optimal paths. Therefore, for each part, we can first obtain the starting point and endpoint of the path defined as $S(x_0, y_0, z_0)$ and $E(x_N, y_N, z_N)$, use $flag(i)$ to mark the validity of the current position and the position to be visited, initialize it to zero, and then update the status

[5] In this study, the drone used for the simulation experiments is represented as a single particle, serving solely for the purpose of the simulation.

through checking, and use $path(i)$ to record the current path, starting from the starting point. At the same time, some movable points are further defined to determine all possible movable points projected from the current position $pre(x_i, y_i, z_i)$ as the choice of the path. The formula is:

$$nextN(i) = pre + direction \tag{6}$$

Where $nextN(i)$ represents the possible movable points of the next position, which is based on the current location $pre(x_i, y_i, z_i)$ and next step $direction$. $direction$ is a matrix containing multiple directional vectors, with each row representing a possible movement direction. This matrix encompasses all basic movement directions the UAV can choose from during the path search process. We have selected six basic directions, forward, backward, left, right, up, and down, to ensure that the UAV can adjust its position flexibly in three-dimensional space.

The definition of the direction matrix is as follows:

$$direction = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \tag{7}$$

In this matrix, the first row, (1,0,0), represents moving one unit in the positive x-axis direction, which means UVA "goes forward.". The second row, (-1,0,0), represents moving one unit in the negative x-axis direction, which means UVA "goes backward." The third row, (0,1,0), represents moving one unit in the positive y-axis direction, which means UVA "goes right" The fourth row, (0,-1,0), represents moving one unit in the negative y-axis direction which means UVA "goes left." The fifth row, (0,0,1), represents moving one unit in the positive z-axis direction, which means UVA "goes up," and the sixth row, (0,0,-1), represents moving one unit in the negative z-axis direction, which means UVA "goes down."

This design allows the UAV to adjust its position in six basic directions in three-dimensional space and ensures flexibility in path searching. In the path planning algorithm, all possible candidate positions for the next step can be generated by adding each directional vector to the current point. Therefore, when selecting the next movement, the UAV will choose the optimal direction based on the priority of these candidate positions and their distance to the target, gradually approaching the target.

Meanwhile, to ensure that the UAV does not attempt to move beyond the defined environment boundary, we will eliminate some out-of-bounds points to ensure that the UAV's actions are within the feasible range, as shown in equation (8).

$$flag(i) = \begin{cases} 1, (x_i \le 0 \ \cup \ x_i > M_x) \cup (y_i \le 0 \ \cup \ y_i > M_y) \cup (z_i \le 0 \ \cup \ z_i > M_z) \\ 0, \quad other \end{cases} \tag{8}$$

Where M_x, M_y, and M_z are the boundary values. The boundary set in this paper is $[300 \times 300 \times 200]$. If the point exceeds the environment boundary, $flag(i)$ is set to 1, indicating that the point is invalid; otherwise, it is 0. This step is the core of path selection, involving the calculation of the distance from the current point to the starting point and endpoint, combined with the priority index of the current point to determine the best movable point, as shown in equation 9:

$$\begin{cases} D_1(i) = |nextN(i) - S| = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \\ D_2(i) = |nextN(i) - E| = \sqrt{(x_i - x_N)^2 + (y_i - y_N)^2 + (z_i - z_N)^2} \\ pri(i) = Status(nextN(i)) \end{cases} \tag{9}$$

Calculate the Euclidean distance from each valid movable point to the starting point S and the endpoint E, denoted as $D_1(i)$ and $D_2(i)$, respectively, and set the priority $pri(i)$ to the value at the corresponding position in the state array x.

Then select one of the best movable points $Norm(x_i, y_i, z_i)$, and the calculation of its comprehensive objective function $Score(i)$ is shown in equation (10):

$$Score(i) = (D_1(i) + D_2(i)) \cdot \sqrt{pri(i)} \tag{10}$$

By finding the point with the lowest score, update the current location $pre(i)$ and $path(i)$ to ensure the accuracy of path recording and dynamic adjustment.

Finally, it calculates the total length D of the entire path and set it as the fitness function:

$$fitness(i) = D = \sum_{i=1}^{N-1} |path(i) - path(i+1)| \tag{11}$$

By accumulating the distances between all path points, obtaining the total length is a reasonable way to calculate the fitness. The smaller the total path length, the higher the fitness, which is suitable for the planning goal of the UAV's three-dimensional path.

3. Improvement of the Parrot Optimization (PO) Algorithm

Parrot optimization is a new type of meta-heuristic optimization algorithm inspired by the behavior of parrots in nature. The design inspiration for this algorithm comes from the social behavior, foraging strategies, and adaptability to the environment of parrots. The PO algorithm solves complex optimization problems by simulating the behavioral patterns of parrots in their natural environment, such as searching for food, group collaboration, and social learning.

3.1 Traditional Parrot Optimization

Each parrot represents a possible path in the UAV three-dimensional path planning problem. In this problem, we randomly select three scattered points of a path:

Population Initialization

For the proposed PO optimization, we set the population size N, the maximum number of iterations. Max_{iter}, and the search space boundaries lb (lower bound) and ub (upper bound). The initialization formula is expressed as:

$$X_i^0 = lb + rand(0,1) \cdot (ub - lb) \tag{12}$$

Where $rand(0,1)$ represents a random number between [0, 1], and X_i^0 represents the position of the i-th pigeon in the initial stage. We set the population size to 30 and the maximum number of iterations to 50.

Foraging Behavior

In foraging behavior, they mainly estimate the approximate location of food by observing the location or considering the location of their owner and then flying to their respective locations. The position movement follows the following equation:

$$X_i^{t+1} = (X_i^t - X_{best}) \cdot Levy(\dim) + rand(0,1) \cdot \left(1 - \frac{t}{Max_{iter}}\right)^{\frac{2t}{Max_{iter}}} \cdot X_{mean}^t \tag{13}$$

Where X_i^t represents the current position while X_i^{t+1} means the updated next position. X_{mean}^t represents the average position of the current population, and $Levy(\dim)$ represents the Levy distribution used to describe the flight of parrots. X_{best} represents the best position searched from initialization to the current iteration and also represents the current position of the owner. t represents the current iteration number, $(X_i^t - X_{best}) \cdot Levy(\dim)$ represents the movement based on the relative position to the owner, and $rand(0,1) \cdot \left(1 - \frac{t}{Max_{iter}}\right)^{\frac{2t}{Max_{iter}}} \cdot X_{mean}^t$ represents observing the overall position of the population to lock the direction of food further. We set the maximum $Levy$ flight strategy expression as:

$$\begin{cases} Levy(\dim) = \frac{\mu \cdot \sigma}{\frac{1}{|v|^{\beta}}} \\ \mu \sim N(0, \dim) \\ v \sim N(0, dim) \\ \sigma = \left(\frac{\Gamma(1+\gamma) \cdot \sin\left(\frac{\pi\gamma}{2}\right)}{\Gamma\left(\frac{1+\gamma}{2}\right) \cdot \gamma \cdot 2^{\frac{1+\gamma}{2}}}\right)^{\gamma+1} \end{cases} \tag{14}$$

Where γ is the parameter of the Levy flight strategy, set to 1.5; Γ is the gamma function;

Staying Behavior

Parrots are highly social creatures, and their staying behavior mainly includes suddenly flying to any part of the owner's body and staying there for a period of time.

$$X_i^{t+1} = X_i^t + X_{best} \cdot Levy(\dim) + rand(0,1) \cdot ones(1, dim) \tag{15}$$

Where $ones(1,\dim)$ represents an all-1 vector of dimension dim, $X_{best}\cdot Levy(\dim)$ represents the process of flying to the owner, and $rand(0,1)\cdot ones(1,\dim)$ represents randomly staying on a part of the owner's body.

Communicating Behavior

Parrots are characterized by close communication within the group, including flying towards and away from the flock. It is assumed that the probability of these two behaviors occurring is equal, and using the average position of the current population to symbolize the center of the group:

$$X_i^{t+1}=\begin{cases}0.2\cdot rand(0,1)\cdot\left(1-\frac{t}{Max_{iter}}\right)^{\frac{2t}{Max_{iter}}}\cdot(X_i^t-X_{mean}^t), P\le 0.5\\ 0.2\cdot rand(0,1)\cdot\exp\left(-\frac{t}{rand(0,1)\cdot Max_{iter}}\right) \quad , P>0.5\end{cases} \tag{16}$$

Where P is the switching factor used to randomly determine the probability value of individual parrot behavior, when the P value is less than or equal to 0.5, the individual parrot will execute the first behavior, i.e., fly towards the center of the flock for communication; when the P value is greater than 0.5, the individual parrot will execute the other behavior, i.e., immediately fly away from the center of the flock after communication. It plays the role of random decision-making.

Fear of Strangers' Behavior

Parrots show a natural fear of strangers, keeping their distance from unfamiliar individuals and seeking a safe distance from their owners:

$$X_i^{t+1}=X_i^t+rand(0,1)\cdot\cos\left(0.5\pi\cdot\frac{t}{Max_{iter}}\right)\cdot(X_{best}-X_i^t)-\cos\left(rand(0,1)\cdot\pi\right)\cdot\left(\frac{t}{Max_{iter}}\right)^{\frac{2t}{Max_{iter}}}\cdot(X_i^t-X_{best}) \tag{17}$$

Where $rand(0,1)\cdot\cos\left(0.5\pi\cdot\frac{t}{Max_{iter}}\right)\cdot(X_{best}-X_i^t)$ shows the process of reorienting and flying towards the owner, while $\cos\left(rand(0,1)\cdot\pi\right)\cdot\left(\frac{t}{Max_{iter}}\right)^{\frac{2t}{Max_{iter}}}\cdot(X_i^t-X_{best})$ shows the movement process of moving away from strangers.

The algorithm flow chart of parrot optimization is shown in Figure 3.

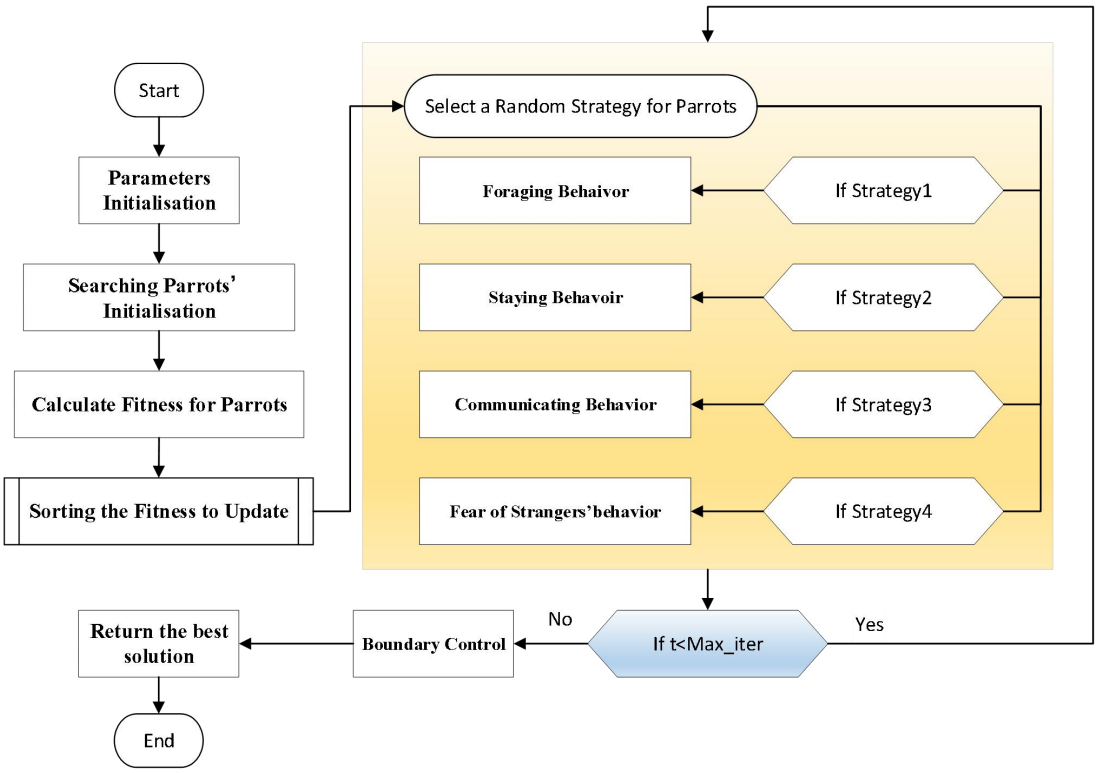


Figure 3: Flowchart of the parrot optimization algorithm

3.2 Improved Parrot Optimization

Since the processing of the UAV three-dimensional path planning problem is more complex than common path planning problems, to make the parrot optimization algorithm equally robust in UAV three-dimensional path planning, we improve the traditional parrot optimization. Improvements are made mainly from population initialization, parrot behavior, and solution space search to enhance the flexibility and efficiency of the algorithm.

3.2.1 SPM Chaotic Mapping

To improve the accuracy of the parrot optimization algorithm, we are considering introducing spatial pyramid matching (SPM) chaotic mapping for optimization. SPM chaotic mapping is an optimization method based on chaos theory, which has attracted attention due to its high exploratory and global search capabilities. Compared with traditional optimization algorithms, SPM can effectively avoid local optimal solutions in complex search spaces and improve the efficiency and accuracy of problem-solving. The chaotic dynamic characteristics make the search process more flexible and changeable, thus increasing the possibility of finding optimized solutions in nonlinear and complex environments. By introducing SPM chaotic mapping, researchers can better handle high-dimensional and highly complex problems and promote the development of various optimization tasks.

When initializing the parrot optimization population, using SPM mapping for optimization can make the population distribution in space more uniform, thus compensating for the shortcomings of random initialization.

$$x_{n+1} = \begin{cases} mod\left[\frac{x_n}{\eta} + \mu \cdot sin\left(\pi \cdot x_n + r\right)\right] & ,0 \le x_n < \eta \\ mod\left[\frac{x_n/\eta}{0.5-\eta} + \mu \cdot sin\left(\pi \cdot x_n + r\right)\right] & ,\eta \le x_n < 0.5 \\ mod[\frac{1-x_n/\eta}{0.5-\eta} + \mu \cdot sin\left(\pi \cdot (1-x_n) + r\right)], 0.5 \le x_n < 1-\eta \\ mod[\frac{1-x_n}{\eta} + \mu \cdot sin\left(\pi \cdot (1-x_n) + r\right)] & ,1-\eta \le x_n < 1 \end{cases} \tag{18}$$

Where mod is the modulo operation; when $\eta \in (0,1), \mu \in (0,1)$, the system is in a chaotic state; r is a random number between 0-1.

The introduction of the SPM chaotic mapping enhances the randomness of the parrot optimization algorithm, thereby improving its global search capability. It effectively helps to avoid local optima and facilitates better path planning.

3.2.2 Adaptive Switching Factor

In parrot optimization, the behavior of individuals (such as foraging, staying, communicating, and fear of strangers) is determined by random selection. Fixed probability selection may lead to insufficient algorithm exploration ability in some cases, especially when the search space is large or complex.

To improve the global search ability of the algorithm in the early stage, we introduce an adaptive switching factor H in the communication stage of parrots, which is dynamically adjusted according to the current number of iterations. The formula is as follows:

$$H = rand(0,1) \cdot \frac{Max_{iter} - i}{Max_{iter}} \tag{19}$$

As the number of iterations increases, the value of $\frac{Max_{iter} - i}{Max_{iter}}$ gradually decreases, which means that in the early stage, the value of H is larger, and individuals are more inclined to explore, while in the later stage, the value of H is smaller, and individuals are more inclined to exploit the known optimal solution.

Through this improvement, the algorithm can better explore the solution space, and in the later stage, it can better utilize the existing information, thereby improving the convergence speed and solution quality and enhancing the flexibility of the algorithm.

3.2.3 Hybrid Cauchy and Gaussian Mutation

In heuristic optimization algorithms, mutation operation is essential to introduce diversity and avoid local optima, aiming to explore a wider search space. Traditional mutation methods may

lead to instability in the search process. By combining the Cauchy distribution's and Gaussian distribution's mutation strategy, the algorithm can adopt different mutation methods at different stages. The Cauchy distribution has a larger tail, which can produce larger mutation amplitudes, while the Gaussian distribution provides smaller, local mutations.

Cauchy mutation uses the Cauchy distribution to generate random numbers with a heavier tail, allowing it to explore more widely in the search space. The formula for Cauchy mutation is:

$$V_{Cauchy} = X_i^t + \gamma \cdot tan\,(\pi \cdot (U(0,1) - 0.5)) \tag{20}$$

Where V_{Cauchy} is the new position after the Cauchy mutation, X_i^t is the current position of the i-th individual in the t-th iteration, γ is the parameter that controls the mutation amplitude, and $U(0,1)$ is a uniformly distributed random number.

Gaussian mutation uses a normal distribution to generate random numbers, which is suitable for fine-tuning around the current solution. The formula for Gaussian mutation is:

$$V_{Gaussian} = X_i^t + \sigma \cdot N(0,1) \tag{21}$$

Where $V_{Gaussian}$ is the new position after Gaussian mutation, σ is the standard deviation, which controls the range of mutation, and $N(0,1)$ is a standard normally distributed random number.

We combine Cauchy and Gaussian mutations, and the formula is expressed as:

$$V_{mixed} = p \cdot V_{Cauchy} + (1-p) \cdot V_{Gaussian} \tag{22}$$

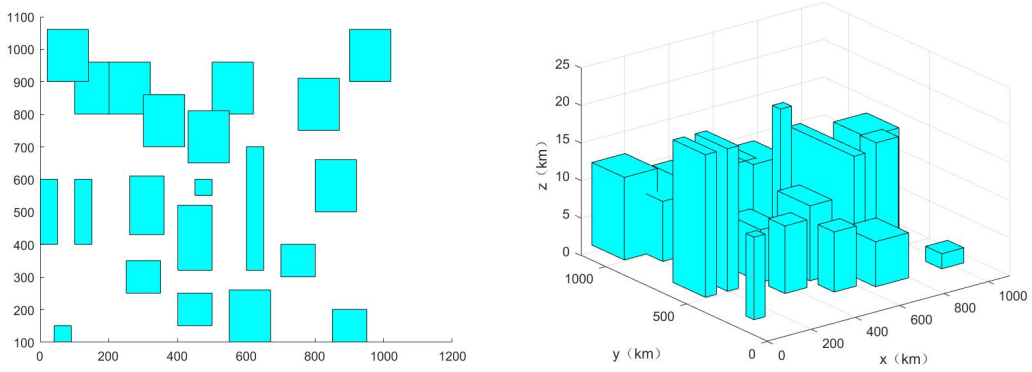
Where V_{mixed} is the final mutation result, p is a weight factor used to control the proportion of Cauchy and Gaussian mutations.

By mixing Cauchy and Gaussian mutations, larger mutations can be introduced in the early stage of the search to help the algorithm jump out of local optima, while smaller mutations can be used in the later stage of the search to fine-tune the solution, allowing the algorithm to achieve a better balance between global search and local search. Cauchy mutation provides strong global search capability, while Gaussian mutation ensures that fine-tuning can be performed after finding potential optimal solutions. By using different mutation strategies at various stages, the algorithm can converge faster to the global optimal solution, more effectively avoiding local optimal solutions while maintaining a specific diversity and avoiding premature convergence, thus increasing the probability of finding the global optimal solution.

4. Experiments and Discussion

4.1 Model Comparison

IPO with Simulated Annealing (SA), Siqi Lin. in 2024 proposed Improved Artificial Bee Colony optimization(ABC)[23], Zebiao Shan et al. in 2023 proposed Improved Whale Optimization Algorithm, IWOA)[24], and Improved Sparrow Search Algorithm (ISSA)[25] newly proposed by Jinwei Zhang et al. in 2024 as a comparison. To improve the efficiency of the experiment, we simulated a three-dimensional environment map. The visualization of the three-dimensional map is shown in the figure.



A B

Figure 4: Simulation environment (A for two-dimensional, B for three-dimensional)

MATLAB is used as a simulation tool for testing. The experimental results are statistically

analyzed, and the comparison results are shown in Table 1 and Figure 5.

Table 1. Compared Results

Evaluation indicators	parameters	IPO	SA	IWOA	SSA	ABC
Fitness	Best	34.58432	35.70531	35.90221	34.15483	34.92142
	Average	35.67941	36.74823	39.08233	37.72813	36.92712
	Slowest	1018.213	1033.892	1181.211	1173.021	1033.212
convergence	Average	890.3123	901.2394	902.3213	891.2231	895.3213
time	Fastest	838.0284	854.8190	855.8576	859.9052	840.6450

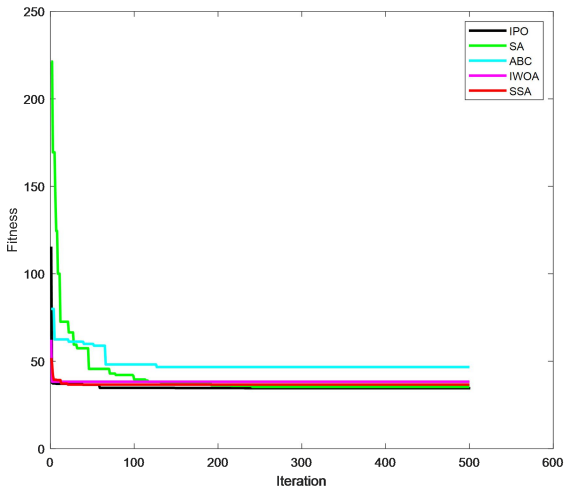


Figure 5: Iteration curve at 500 iterations

The iteration curve comparison graph shows the fitness values change during the iteration process for five optimization algorithms (IPO, SA, ABC, WOA, and ISSA). The number of iterations is represented by the X-axis of the diagram, and the Y-axis represents the fitness value. It can be clearly observed that IPO has a significant decrease in the initial phase with a rapid decrease in fitness, indicating that it excels in its ability to search for the optimal solution and can find a better solution within a shorter number of iterations. In comparison, the adaptation curves of IWOA and SSA are slightly flatter, and although the final results can converge better, the decrease rate is not as significant as that of the improved parrot optimization algorithm in the initial stage.

The performance of SA is even slower, with a smoother trend of improvement in fitness, suggesting that it may have adopted a more cautious strategy in exploring the optimal solution, which in turn may have led to a relatively slower overall convergence rate. Finally, ABC's fitness curve stays in a high fitness range with a limited decrease, suggesting that it may have fallen into a local optimum during the search process, which ultimately fails to improve the results significantly.

In summary, IPO exhibits the best convergence speed and fitness performance in most cases, while IWOA and SSA perform well but slightly less than IPO, and ABC and SA are relatively weak in convergence ability and effectiveness.

Based on this, we use IPO for UAV three-dimensional path planning, and the planned path is shown in Figure 6.

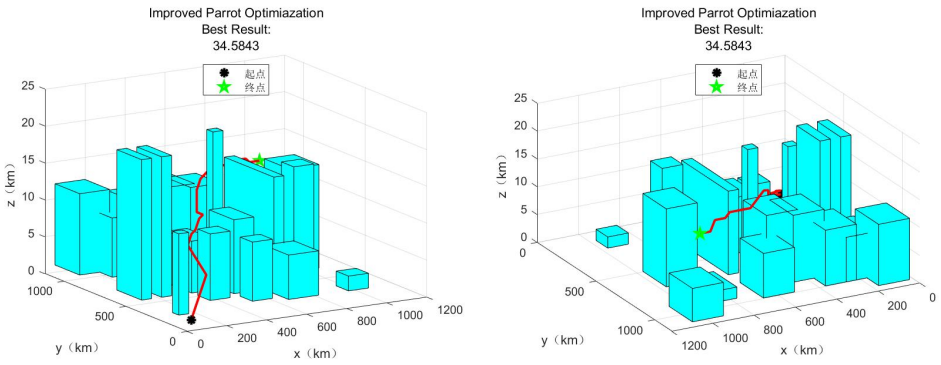


Figure 6: Roadmap of the Best Result of the IPO

4.2 Simulation Experiments of Three-Dimensional UAV Path Planning Based on the BISU Map

Therefore, based on the above content, we selected two mid-points and four endpoints for our simulation experiments. The markings are shown in the Figures below. Numbers represent mid-points, and letters represent target points.

Table 2: Parameter values for the BISU map

	Parameter	Value
Map	Physical Units	1
	Map Units	2
	Physical Size	500*350*50
	Experimental Area	300*300*20
Algorithm	Population Size	30
	Iteration Count	50

Table 3: Parameter values for BISU flight experiments

	Code	Actual Meaning	Coordinates
mid-point	1	Fengwei Canteen	[208，234，2]
	2	Xiangyu Canteen	[126，238，2]
End Point	A	Residential Building	[220，50，2]
	B	Building 7	[20，200，2]
	C	Building 3	[184，106，2]
	D	Building 1	[76，84，2]

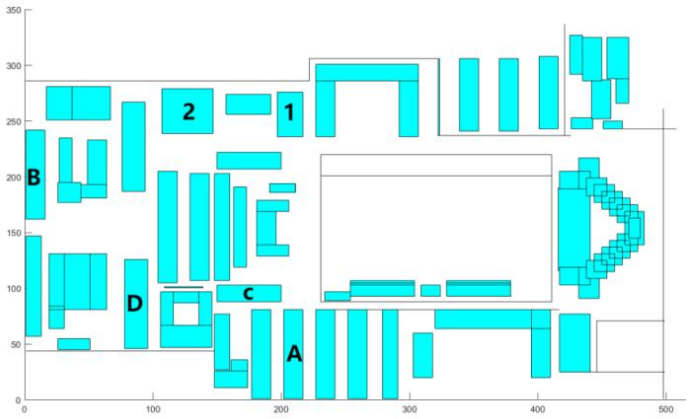


Figure 7: Two-dimensional distribution of coordinates

In the previous section, we discussed considering the traveling salesman problem as a realistic scenario for drone transportation. To reflect the situation where drones often need to visit multiple locations in a single trip, we introduced the concept of transfer points. In our experimental design, the setup of transfer points holds practical significance as it connects various campus facilities with student dormitories at BISU. Specifically, the transfer points are located at the university's dining halls, while the endpoints are the dormitory buildings on campus, and the drones' starting point is the comprehensive living area of the university. In the future, when

residential buildings have procurement needs, logistics drones can deliver the required items directly to the dormitories. For dining needs, items can first be transferred to the dining halls via drones before being delivered to the dormitories, enabling efficient and convenient delivery of goods and meals.

Therefore, it can be seen that the experiment simulates the flight path of the UAV after flying to the canteen and then flying to the residential building and three dormitory buildings. There are a total of 2*4=8 paths. After inputting the above coordinates in the relevant positions in our code, we can successively obtain our experimental results as shown in the following figures[6]:

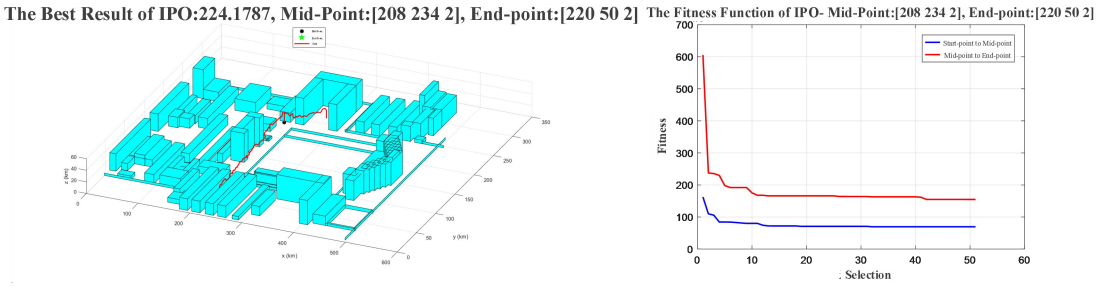


Figure 8: Fengwei Canteen to Residential Building (A is the flight path, B is the fitness function curve)

The fitness function curve illustrates the convergence performance of the algorithm for each path during the iterative process. By introducing SPM chaotic mapping, adaptive switching factors, and a hybrid Cauchy/Gaussian mutation strategy, our method significantly enhances the optimization accuracy of path planning. Compared with traditional algorithms, the IPO algorithm can find shorter flight paths, thereby effectively reducing drone flight costs.

At the same time, the fitness function curve also reveals that the IPO algorithm demonstrates rapid convergence in the early stages of iteration, requiring only a few iterations to achieve an optimal solution. It is attributed to the global search advantages of SPM chaotic mapping and the dynamic adjustment capabilities provided by the adaptive switching factor. Additionally, the paths generated by the IPO algorithm in three-dimensional space are more optimized, significantly reducing the need for sharp turns and frequent attitude adjustments, contributing to improved flight stability and safety.

Besides these, In Section 2.3, we set the minimum flight altitude to 8 meters, but no upper limit was specified for the maximum flight altitude. Therefore, our flight experiment results also include situations where the drone flew over buildings. Our experiment contains those kinds of results, which can be found in Appendix Figures 9, 11, 13, and 15. Moreover, based on the results from 9, 11, 13, and 15, we can see that the optimal flight paths to Building 7 and Building 1 are flying over the buildings obstructing their route.

In conclusion, the 3D path planning simulation experiments based on the BISU map validate the effectiveness and robustness of the IPO algorithm in drone path planning. By incorporating multiple improvement strategies, the algorithm significantly enhances optimization accuracy and convergence speed and improves path smoothness, providing an efficient solution for autonomous drone navigation in complex environments.

5. Conclusion

This paper addresses the problem of 3D path planning for unmanned aerial vehicles (UAVs) in complex urban environments and proposes an Improved Parrot Optimizer (IPO) algorithm. By introducing SPM chaotic mapping, adaptive switching factors, and a hybrid Cauchy and Gaussian mutation strategy, the algorithm's global search capability and convergence speed are significantly enhanced. Combining the MATLAB simulation platform, a complete 3D path planning solution framework is constructed, and simulation experiments verify that the improved algorithm achieves

[6] For the sake of readability and formatting, we present only one example of the results here. For detailed results, please click on "Figure" to navigate to Appendix 1.

notable improvements in optimization accuracy, convergence speed, and path smoothness.

Furthermore, this paper conducts detailed 3D environment modeling of the Beijing International Studies University (BISU) campus, where the authors are located, and performs path planning simulation experiments based on this model. The experimental results further validate the application potential of the improved IPO algorithm in complex real-world environments. However, there are still some limitations to the research in this paper. Firstly, exploring the application of the improved IPO algorithm on parallel computing platforms can be considered to enhance the algorithm's processing capability and ability to adapt to larger-scale problems. Secondly, it can be conducted by applying the improved IPO algorithm to multi-UAV collaborative task scenarios to achieve effective coordination and path planning among UAVs. Moreover, considering the dynamic changes in environmental factors, such as wind speed and air currents, the adaptability and robustness of the algorithm under dynamic constraints can be studied. The curve smoothing strategies are employed to make the flight trajectory smoother. Additionally, developing adaptive mechanisms to enable the algorithm to adjust parameters according to the characteristics of the problem automatically can improve the algorithm's versatility and flexibility. Finally, beyond the laboratory environment, actual UAV flight tests can be conducted to verify the effectiveness and feasibility of the algorithm in the real world. The improved IPO algorithm proposed in this paper provides an effective solution for the 3D path planning problem of UAVs and has good prospects for engineering applications. Future research will further explore the optimization and application of the algorithm to promote the application and development of UAV technology in a broader range of fields.

Reference

[1] Research on the Application of Unmanned Aerial Vehicles in Military Reconnaissance [J].Journal of Unmanned Aerial Vehicle Technology，2020，12(3):1-6.

[2] The Application Status and Prospects of Drones in Disaster Relief[J]. Disaster Science，2021，36(2):34-42.

[3] Karaman S， Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research， 2011， 30(7): 846-894.

[4] Roberge V， Tarbouchi M， Labonté G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning[J]. IEEE Transactions on Industrial Informatics， 2012， 9(1): 132-141.

[5] Key Technologies and Challenges in UAV Logistics and Distribution[J]. Logistics Technology and Application，2019，24(5):56-61.

[6] Yan, M., Yuan, H., Xu, J. et al. Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm. EURASIP J. Adv. Signal Process. 2021, 94 (2021). https://doi.org/10.1186/s13634-021-00804-9

[7] Cabreira T M, Brisolara L B, Ferreira Jr P R. Survey on coverage path planning with unmanned aerial vehicles[J]. Drones, 2019, 3(1): 4.

[8] Zhao Y, Zheng Z, Liu Y. Survey on computational-intelligence-based UAV path planning[J]. Knowledge-Based Systems, 2018, 158: 54-64.

[9] Fu Y, Ding M, Zhou C. Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2012, 42(2): 511-526.

[10] Yang X S. A new metaheuristic bat-inspired algorithm[M]//Nature inspired cooperative strategies for optimization. Springer, Berlin, Heidelberg, 2010: 65-74.

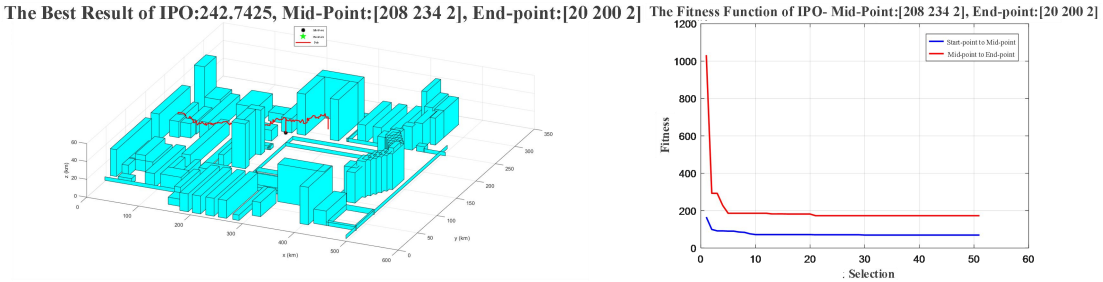
[11] Wang G G, Deb S, Coelho L D S. Elephant herding optimization[C]//2015 3rd International Symposium on Computational and Business Intelligence. IEEE, 2015: 1-5.

[12] Huang H, Tsai Y, Lin J. Optimal route planning for inspection task of multi-UAV system based on improved ACO-ABC algorithm[J]. Aerospace Science and Technology, 2021, 112: 106635.

[13] Alejo D, Cobano J A, Heredia G, et al. Optimal reciprocal collision avoidance with mobile and static obstacles for multi-UAV systems[C]//2014 International Conference on Unmanned Aircraft

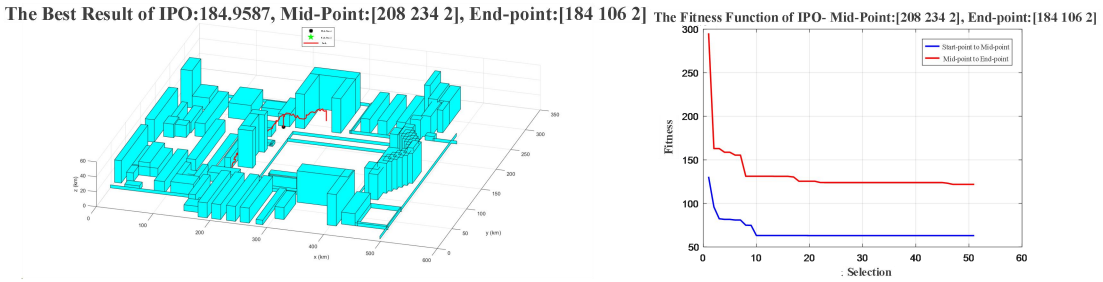
Systems (ICUAS). IEEE, 2014: 1259-1266.
[14] Shao Z, Yan F, Zhou Z, et al. Path planning for multi-UAV formation rendezvous based on distributed cooperative particle swarm optimization[J]. Applied Sciences, 2019, 9(13): 2621.
[15] Zhu W, Duan H. Chaotic predator–prey biogeography-based optimization approach for UCAV path planning[J]. Aerospace Science and Technology, 2014, 32(1): 153-161.
[16] Iacono M, Sgorbissa A. Path following and obstacle avoidance for an autonomous UAV using a depth camera[J]. Robotics and Autonomous Systems, 2018, 106: 38-46.
[17] Qu C, Gai W, Zhang J, et al. A novel hybrid grey wolf optimization algorithm for unmanned aerial vehicle (UAV) path planning[J]. Knowledge-Based Systems, 2020, 194: 105530.
[18] Zhang Y, Wang S, Ji G. A comprehensive survey on particle swarm optimization algorithm and its applications[J]. Mathematical Problems in Engineering, 2015, 2015.
[19] Phung M D, Quach C H, Dinh T H, et al. Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection[J]. Automation in Construction, 2017, 81: 25-33.
[20]Wen S, Zhang Q, Deng J, et al. Three-dimensional path planning for unmanned aerial vehicle based on improved artificial potential field method[J]. Frontiers in Neurorobotics, 2021, 15: 673902.
[21]Gao X, Hou Z, Zhu J, et al. Joint optimization of battery pack capacity and UAV route planning for power line inspection[J]. IEEE Transactions on Power Delivery, 2020, 36(4): 2118-2128.
[22]Hu X, Cheng J, Zhou H, et al. Multiobjective route planning for unmanned aerial vehicle swarms in complex environments[J]. IEEE Access, 2020, 8: 126066-126084.
[23]S. Lin, F. Li, X. Li, K. Jia and X. Zhang, "Improved Artificial Bee Colony Algorithm Based on Multi-Strategy Synthesis for UAV Path Planning," in IEEE Access, vol. 10, pp. 119269-119282, 2022.
[24]Shan, Y. Wang, X. Liu and C. Wei, "Fuzzy Automatic Disturbance Rejection Control of Quadrotor UAV Based on Improved Whale Optimization Algorithm," in IEEE Access, vol. 11, pp. 69117-69130, 2023.
[25]Jinwei Zhang, Xijing Zhu, Jing Li Intelligent Path Planning with an Improved Sparrow Search Algorithm for Workshop UAV Inspection[J]. Sensors 2024, 24(4).

Appendix: The result of the Experiments of Three-Dimensional UAV Path Planning Based on the BISU Map



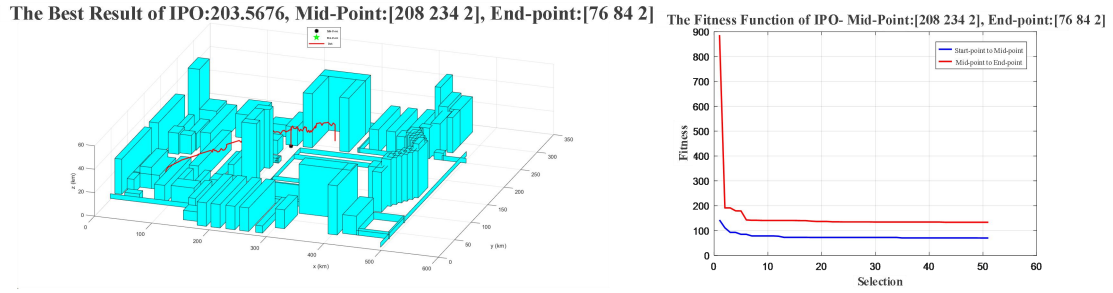
A B

Figure 9: Fengwei Canteen to Building 7 (A is the flight path, B is the fitness function curve)



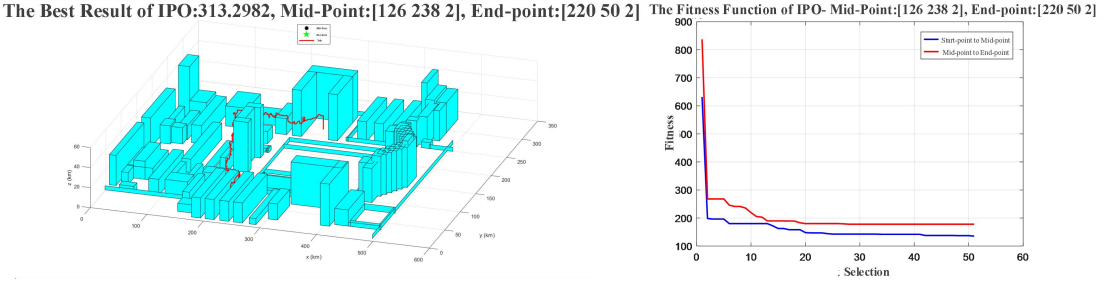
A B

Figure 10: Fengwei Canteen to Building 3 (A is the flight path, B is the fitness function curve)



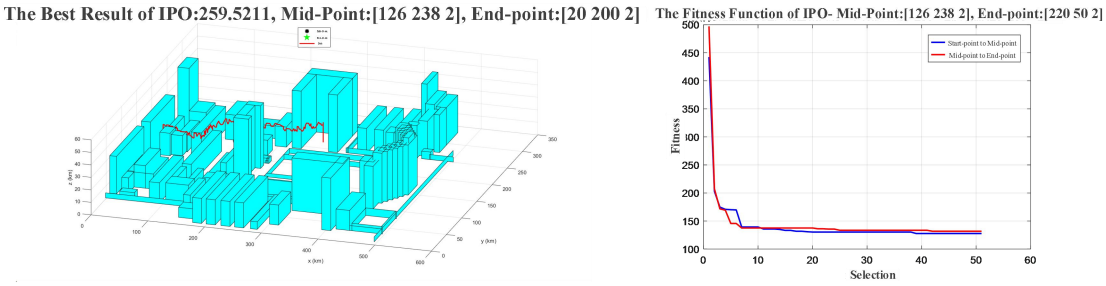
A B

Figure 11: Fengwei Canteen to Building 1 (A is the flight path, B is the fitness function curve)



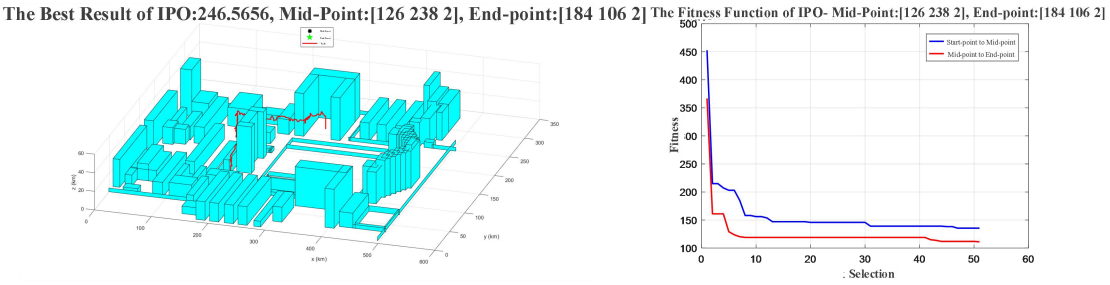
A B

Figure 12: Xiangyu Canteen to Residential Building (A is the flight path, B is the fitness function curve)



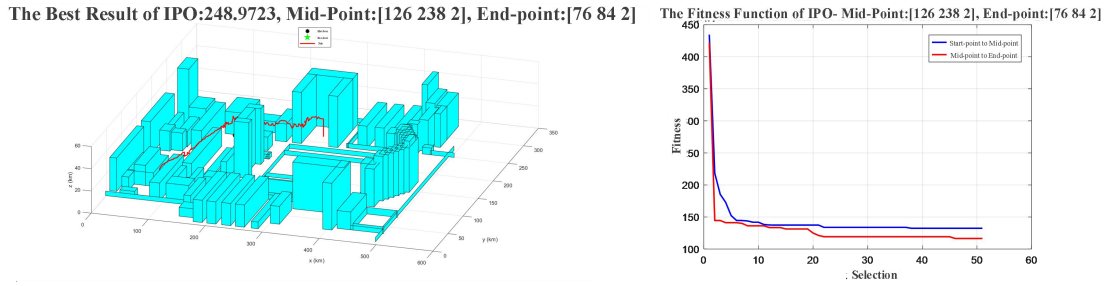
A B

Figure 13: Xiangyu Canteen to Building 7 (A is the flight path, B is the fitness function curve)



A B

Figure 14: Xiangyu Canteen to Building 3 (A is the flight path, B is the fitness function curve)



A B

Figure 15: Xiangyu Canteen → Building 1 (A is the flight path, B is the fitness function curve)