ITC 3/54

Information Technology and Control
Vol. 54 / No. 3/ 2025
pp. 821-843
DOI 10.5755/j01.itc.54.3.41351

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

Received 2025/04/29 | Accepted after revision 2025/07/22

HOW TO CITE: Ma, H., Gao, Y., Zhou, Z. (2025). Three-Dimensional UAV Path Planning in Urban Environments Based on an Improved Parrot Optimization Algorithm. *Information Technology and Control*, 54(3), 821-843. https://doi.org/10.5755/j01.itc.54.3.41351

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

Hongtao Ma
School of Economics, Beijing International Studies University, Beijing 100024, China
Hongtao Ma and Yisheng Gao contributed equally to this work and they are both first authors.

Yisheng Gao
School of Management, Shandong University of Technology, Zibo 255012, Shandong, China
Hongtao Ma and Yisheng Gao contributed equally to this work and they are both first authors.

Zihang Zhou*
School of Economics, Beijing International Studies University, Beijing 100024, China

Corresponding author: 2020221271@stu.bisu.edu.cn

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 Spatial Pyramid Matching (SPM) chaotic mapping, Dynamic adaptive foraging mechanisms, 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 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 modelings of the Beijing International Studies University campus, High-density urban core and Urban areas with rivers and bridges, the IPO 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

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 [9]. 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 [15].

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. Building on this foundation, recent studies have increasingly explored intelligent optimization methods to further enhance the efficiency and robustness of UAV path planning. Bui et al. [4] applied Ant Colony Optimization to cooperative inspection path planning with multiple UAVs, enhancing both path quality and coordination efficiency. Arslan et al. [3] developed sampling-based motion planning algorithms that leverage closed-loop prediction to improve the optimality and reliability of UAV trajectory planning. 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 [11]. To address the inefficiency of conventional algorithms in large-scale, high-dimensional UAV coordination problems, Yan et al. [19] proposed an enhanced particle swarm optimization (PSO) algorithm. Their approach incorporates partial matching crossover and secondary swap mutation—techniques borrowed from genetic algorithms—to effectively tackle random task allocation and two-dimensional path planning in multi-UAV systems. Cabreira et al. [5] 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.

At the same time, swarm intelligence optimization algorithms have received increasing attention in UAV path planning due to their good global search capability and robustness. Zhang et al. [22] 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. [6] proposed a hybrid quantum-behaved particle swarm optimization (PSO) algorithm incorporating phase angle encoding, which effectively enhances solution quality and accelerates convergence. However, the algorithm's performance is still sensitive to parameter tuning, posing a challenge for practical implementation. Zhu and Duan [24] proposed a chaotic predator–prey biogeography-based optimization (BBO) algorithm for unmanned combat aerial vehicle (UCAV) path planning. By integrating chaotic maps and predator–prey dynamics into the BBO framework, the algorithm enhances population diversity and global search capability. Li et al. [9] reviewed the use of UAVs in logistics, emphasizing their benefits in delivery efficiency, cost reduction, and accessibility. They classified applications into last-mile delivery, warehouse management, and emergency response, while noting challenges like airspace regulation, battery limitations, and public acceptance. Therefore, Alejo et al. [1] 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 Robot Operating System framework and conducting various simulation tests, the effectiveness of the algorithm in complex environments is verified. Shao et al. [17] 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. Zhao et al. [23] conducted a survey on computational-intelligence-based UAV path planning, summarizing various heuristic and bio-inspired algorithms, and analyzing their strengths, limitations, and application scenarios. Iacono et al. [8] 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. [14] 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. Yu et al. [20] 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 et al. [13] 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 four strategies: Spatial Pyramid Matching chaotic mapping (SPM), Dynamic adaptive foraging mechanisms, adaptive switching factor, and hybrid Cauchy/Gaussian mutation to address the shortcomings of the standard parrot optimization algorithm. By comparing with today's more advanced optimization algorithms [14, 21, 10, 22, 23, 16], 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

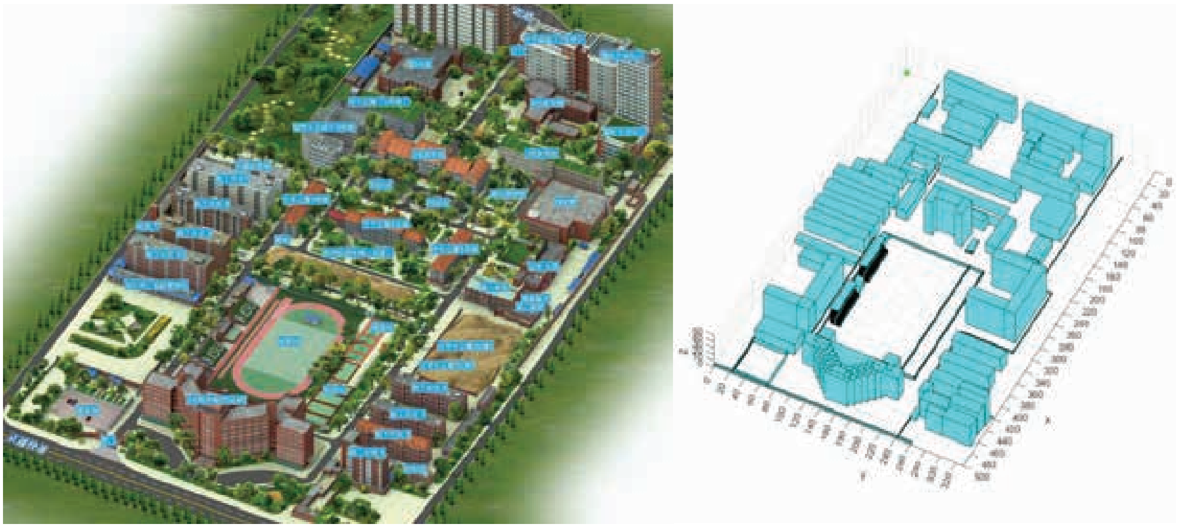
The three-dimensional path planning problem for UAVs in complex environments is essential. The UAV aims to start from a starting point, avoid obstacles, and reach a predetermined endpoint. This paper utilizes an improved parrot optimization algorithm to explore an optimization and evaluation method for 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

Figure 1

Architectural rendering of Beijing International Studies University (a) and modeling result (b).



(a) (b)

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:

As shown in the Figure 1, Figure 1(a) basically models all the buildings on the campus. In the model of Figure 1(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 of the school in reality.

2.2. Map Modeling and Design

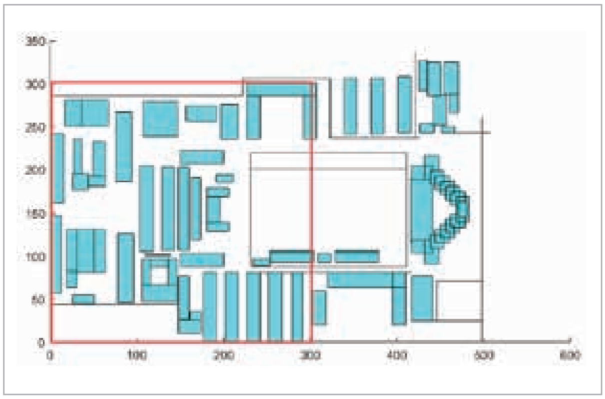
First, we set the boundaries of the virtual map in the three-dimensional space. As shown in Figure 2, we refer to the proportions of length, width, and height in real life and extend from the origin [0,0,0] to constrain the size of the map to [500×50×50]. The three values in the brackets represent the 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 [500×50×50], 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 [300×300×20], which is the range specified in Figure 2. In real life, the main living area of students is 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 [2×2×2]. Dividing the set UAV flight boundary size [300×300×20] by the map unit, we obtain the size of the new map after discretization, which is represented as a three-dimensional matrix of [150×150×10]. 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 and the delineated experimental area.



Then, we design the obstacles in the Figure 2. To represent the initial state of the map (i.e., without obstacles), we define an initial all-zero matrix ($\mathrm{Map}_{\mathrm{Initial}}$) 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 $Obstacle(i) = [X_i, Y_i, Z_i, L_i, W_i, H_i]$. 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 v 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_{end}, y_{end}, z_{end}) of the obstacle, we make the following judgment for all coordinates $map(x, y, z)$ in the map, 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.

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 h_1 is 4.905 meters (where t = 1 second and g = 9.81 m/s^2). 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 is 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 is 1 meter, then the total minimum safe altitude is $h_{\min} = h_1 + h_2 + h_3 = 7.905$m.

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 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:

$$nextN(i) = pre + direction, \qquad (1)$$

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:

$$direction = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \qquad (2)$$

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 (3):

$$\mathrm{flag(i)} = \begin{cases} 1, (x_i \le 0 \cup x_i > M_x) \cup (y_i \le 0 \cup y_i > M_y) \\ 0, \qquad \text{other} \end{cases} \qquad (3)$$
$$\cup (z_i \le 0 \cup z_i > M_z)$$

where M_x, M_y, and M_z are the boundary values. The boundary set in this paper is [300×300×200]. 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 4:

$$\begin{cases} D_1(i) = |\mathrm{nextN}(i) - S| = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \\ D_2(i) = |\mathrm{nextN}(i) - E| = \sqrt{(x_i - x_N)^2 + (y_i - y_N)^2 + (z_i - z_N)^2} \\ \mathrm{pri}(i) = \mathrm{Status}(\mathrm{nextN}(i)) \end{cases}. \quad (4)$$

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 $\mathrm{Score}(i) = (D_1(i) + D_2(i)) \cdot \sqrt{\mathrm{pri}(i)}$. 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 $fitness(i) = \sum_{i=1}^{N-1} | path(i) - path(i+1)$.

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

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. PO 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. Simple Parrot Optimization

Each parrot represents a possible path in the UAV 3D 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:

$$X_i^0 = \mathrm{lb} + \mathrm{rand}(0,1) \cdot (\mathrm{ub} - \mathrm{lb}), \quad (5)$$

where rand(0,1) represents a random number between [0, 1], and x^0_i 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 300.

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 movement follows the equation:

$$X_i^{t+1} = (X_i^t - X_{best}) \cdot Levy(\cdot) + rand(\cdot)\left(1 - \frac{t}{Max_{it}}\right)^{Max_{it}} \cdot X_{mean}^t. \quad (6)$$

where Levy() 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. The Levy flight strategy simulates the parrot's behavioral pattern of foraging, sometimes with small fine searches and sometimes with large jumps of exploration. This part simulates the parrot's exploratory flight based on self-experience (i.e., memory of optimal locations). the maximum Levy flight strategy expression as:

$$\begin{cases} \mathrm{Levy}(\mathrm{dim}) = \frac{\mu \cdot \sigma}{|v|^{\frac{1}{\beta}}} \\ \mu \sim N(0, \mathrm{dim}) \\ v \sim N(0, \mathrm{dim}) \\ \sigma = \left(\frac{\Gamma(1+\gamma) \cdot \sin\left(\frac{\pi\gamma}{2}\right)}{\Gamma\left(\frac{1+\gamma}{2}\right) \cdot \gamma \cdot 2^{\frac{1+\gamma}{2}}}\right)^{\gamma+1} \end{cases}, \quad (7)$$

where ϒ is the parameter of the Levy flight strategy, set to 1.5; Γ is the gamma function. It is characterized by fine searching in a small area most of the time, with an occasional long jump to explore. Here, it mimics the ′step length′ and ′direction′ of parrot flight, so that the parrot does not just fly in a straight line towards the owner, but rather searches in an exploratory, hopping manner in the vicinity of the owner.

$rand(\cdot) \cdot \left(1 - \frac{t}{Max_{it}}\right)^{\frac{2t}{Max_{it}}} \cdot X_{mean}^{t}$ represents observing the overall position of the population to lock the direction of food further, The weighting factor changes dynamically as the number of iterations *t* increases, which makes the population influence largest in the middle of the algorithm iteration and smaller in the beginning and end, which simulates that the parrot prefers to explore on its own at the beginning of the search process; in the middle it refers more to the population information; and at the end it reduces the dependence on the average position and focuses on the localized and fine-grained search when the optimal solution may have been found.

The Foraging Behavior combines the parrot′s individual exploration and social learning strategies, and the next position is the result of a combination of the “tendency to fly to the known population optimum” and the “tendency to fly to the area where the majority of the individuals are located (the mean)”, simulating the process of its intelligent foraging.

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.

$$X_i^{t+1} = X_i^t + X_{best} \cdot Levy(\cdot) + rand(\cdot) \cdot ones(\cdot), \tag{8}$$

where *ones*(·) represents an all-1 vector of dimension, $X_{best} \cdot Levy(\cdot)$ represents the process of flying to the owner, a single jump of the parrot towards the optimal position is simulated by means of the Levy flight operator. And *rand*(·) · *ones*(·) generated a small, dimensionally random displacement, which simulates that the parrot does not land precisely at the “center” of its owner, but stays at a random part of its body. This randomness allows the algorithm to perturb a very small neighborhood of the current optimal solution, which facilitates a fine-grained search and allows it to leapfrog out of a small local optimum. This section describes a process of “big steps to approach + small steps to fine-tune”, in which the parrot first makes a “leap” to reach roughly the location of its owner (the optimal solution), and then makes a small random displacement to determine the final “stopping point”. Then a small random displacement is used to determine the final “stopping point”, which is represented in the algorithm as a fast approach to the optimal solution and a fine-grained local search in its vicinity.

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:

$$X_i^{t+1} = \begin{cases} 0.2 \cdot \mathrm{rand}(0,1) \cdot \left(1 - \frac{t}{\mathrm{Max_{iter}}}\right)^{\frac{2t}{\mathrm{Max_{iter}}}} \cdot (X_i^t - X_{\mathrm{mean}}^t), P \le 0.5 \\ 0.2 \cdot \mathrm{rand}(0,1) \cdot \exp\left(-\frac{t}{\mathrm{rand}(0,1) \cdot \mathrm{Max_{iter}}}\right) \quad , P > 0.5 \end{cases} . \tag{9}$$

When P 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, the whole equation represents that the position of an individual parrot will be adjusted according to its relative vector to the center of the group. When P 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, the exp function makes this part rapidly smaller as the number of iterations increases, in effect giving the parrot a random perturbation that decays over time, simulating the behavior of “ending the exchange and leaving”, not by flying in a super-specific direction, but by a small random “jumping out of the way Instead of flying in a super-specific direction, a small random “jump away” is used to explore new possibilities and avoid having all individuals clustered in the center. 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:

$$X_i^{t+1} = X_i^t + rand(\cdot)\cdot\cos\left(0.5\pi\cdot\frac{t}{Max_{it}}\right)\cdot\left(X_{best} - X_i^t\right) - \cos\left(rand(\cdot)\cdot\pi\right)\cdot\left(\frac{t}{Max_{it}}\right)^{\frac{2t}{Max_{it}}}\cdot\left(X_i^t - X_{best}\right), \quad (10)$$

where the first half of the formula shows the process of reorienting and flying towards the owner, $X_{best} - X_i^t$ is a vector pointing from the parrot's current position to the owner's position, which clearly represents the direction of "flying to the owner"; $\cos\left(0.5\pi\cdot\frac{t}{Max_{it}}\right)$ is a weight decreasing from 1 to 0 over time, which simulates that the parrot's fear will slowly decrease over time, thus the urge to fly to the owner will become weaker. The parrot's fear decreases slowly over time, so the urge to fly to the owner becomes weaker. while the second half of the equation shows the movement process of moving away from strangers, where $X_i^t - X_{best}$ represents the direction "away from the owner"; $\cos(rand(\cdot)\cdot\pi)$ produces a random number between -1 and 1, simulating the parrot's irregularities and hesitations in fear, which may suddenly fly a little bit further away from the owner, or may accelerate more towards the owner because of two negative numbers. may fly towards the owner more quickly because of the two negative numbers thus.

The flow chart of PO is shown in Figure 3.

3.2. Improved Parrot Optimization

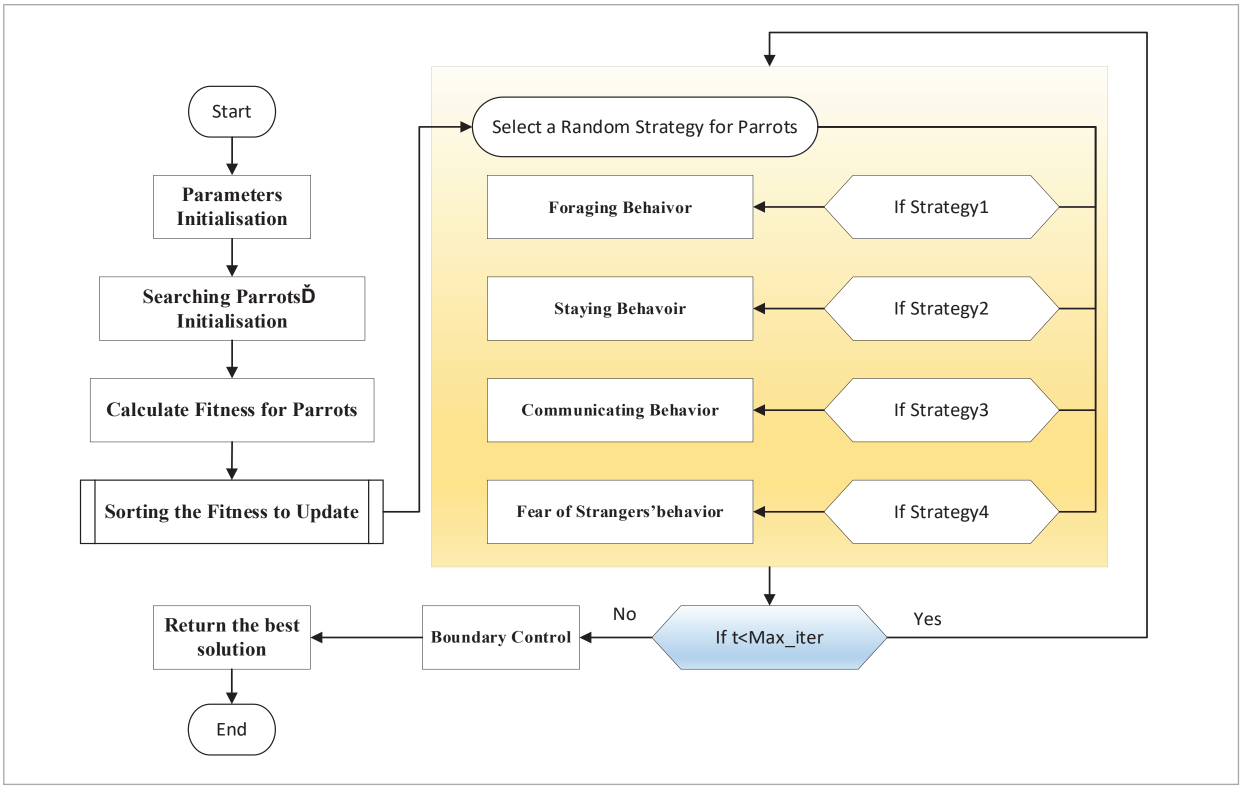
Since the processing of the UAV 3D path planning problem is more complex than common path planning problems, to make the PO equally robust in UAV 3D path planning, we improve the original PO.

3.2.1. SPM Chaotic Mapping

To improve the accuracy of the PO, 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. SPM chaotic mapping can effectively avoid local optimal solutions in complex search spaces and improve the efficiency and accuracy of problem-solving. The chaotic dy-

Figure 3

Flowchart of the parrot optimization.


Start
Parameters Initialisation
Searching ParrotsĎ Initialisation
Calculate Fitness for Parrots
Sorting the Fitness to Update
Select a Random Strategy for Parrots
Foraging Behaivor
If Strategy1
Staying Behavoir
If Strategy2
Communicating Behavior
If Strategy3
Fear of Strangers'behavior
If Strategy4
No
If t<Max_iter
Yes
Boundary Control
Return the best solution
End

namic 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, we can better handle high-dimensional and highly complex problems and promote the development of various optimization tasks.

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

$$x_{n+1}=\begin{cases}\bmod\left[\frac{x_n}{\eta}+\mu\cdot\sin(\pi\cdot x_n+r)\right] & ,0\le x_n<\eta\\ \bmod\left[\frac{x_n/\eta}{0.5-\eta}+\mu\cdot\sin(\pi\cdot x_n+r)\right] & ,\eta\le x_n<0.5\\ \bmod\left[\frac{1-x_n/\eta}{0.5-\eta}+\mu\cdot\sin(\pi\cdot(1-x_n)+r)\right] & ,0.5\le x_n<1-\eta\\ \bmod\left[\frac{1-x_n}{\eta}+\mu\cdot\sin(\pi\cdot(1-x_n)+r)\right] & ,1-\eta\le x_n<1\end{cases}. \tag{11}$$

When $\eta \epsilon (0,1)$, $\mu \epsilon (0,1)$ 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 PO, 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 PO, the behavior of individuals (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:

$$H = rand(0,1)\cdot\frac{Max_{iter}-i}{Max_{iter}}. \tag{12}$$

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. Dynamic Adaptive Foraging Mechanisms

Seeing the formula for Foraging Behavior in the original PO (see 6), it can be seen that this behavior is mainly guided by X_{best}, and in the early stage of the algorithm, if X_{best} happens to be a locally optimal solution at the current stage, then the entire population prematurely approaches it, which will greatly increase the risk of falling into a local optimum, weakening the algorithm's ability to explore the whole world; and the formula uses X^{t}_{mean} for calculation, which may cause the limitation of partial information guidance, in the swarm intelligence optimization algorithm, the average position of the whole population X_{mean} is often chosen as a more meaningful guidance information, because it represents the central tendency and collective wisdom of the whole population.

In order to solve this problem, we introduce a dynamic adaptive foraging guidance strategy, the core idea of the strategy is: in different stages of the algorithm iteration, dynamically adjust the Foraging Behavior of the "learning object", for the early iteration, the parrot should learn more from the average position X_{mean}, which is equivalent to the whole population to explore the center of the region. This is equivalent to exploring the center of the region where the whole population is doing well, which is beneficial to the global search and avoids falling into a local extreme point too early. As the optimal solution gets closer to the global optimal solution, the parrot should learn more from the global optimal position to accelerate the convergence and perform finer local search. To this end, we achieve this smooth transition by a nonlinearly decreasing weight factor w, the formula is as follows:

$$w = cos\left(\frac{i}{Max_{iter}}\cdot\frac{\pi}{2}\right). \tag{13}$$

The weights smoothly transition from 1 to 0 as the number of iterations i increases. A dynamic bootstrap targeting mechanism is then constructed for the parrot's foraging behavior: $guidance = w\cdot X_{meaan} + (1-w)\cdot X_{best}$, a modified Foraging Behavior updating equation:

$$X_i^{t+1} = X_i^t + Levy(\cdot) \cdot (guidanc\_target - X_i^t). \quad (14)$$

Each parrot performs a levy flight based on its current position in a direction jointly determined by the dynamic guidance mechanism and the current position it is in.

3.2.4. 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:

$$V_{Cauchy} = X_i^t + \gamma \cdot \tan(\pi \cdot (U(0,1) - 0.5)), \quad (15)$$

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, Υ is the parameter that controls the mutation amplitude, and U(0,1) is a unidformly 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:

$$V_{Gaussian} = X_i^t + \sigma \cdot N(0,1), \quad (16)$$

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:

$$V_{mixed} = p \cdot V_{Cauchy} + (1 - p) \cdot V_{Gaussian}, \quad (17)$$

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) [10], Shan et al. in 2023 proposed Improved Whale Optimization Algorithm, IWOA) [16], and Improved Sparrow Search Algorithm (SSA) [21] newly proposed by Zhang et al. in 2024 as a comparison.

The most fundamental difference between IPO and SA is that one is a "group" and the other is an "individual", first of all, the search theme is different, IPO is a population-based algorithm, which maintains a population of N parrots at the same time, and the individuals interact with each other by imitating the optimal individual and referring to the mean value of the population. Individuals interact with each other by imitating the optimum, referring to the mean value of the population, etc., and jointly push the population to evolve to more regions, while SA is an algorithm based on a single solution, which maintains only one current solution at any moment, and the search process is the trajectory of a single solution in the solution space; secondly, the mechanisms of escaping from the local optimum of the two are also different, as IPO relies on the diversity of the population and the perturbations introduced by the behaviors of Levy's flights, such as long-distance jumps, to explore different regions, and thus jump out of the local optimum of the two. to explore different regions and thus jump out of the local optimum, while the core mechanism of SA is probabilistic acceptance of poor solutions, according to a gradually decreasing temperature T, SA will accept a worse solution than the current one with a certain probability. At high temperatures it is more likely to accept poor solutions for extensive exploration, while at low temperatures it tends to accept only superior solutions for local convergence.

Although IPO and ABC and SSA are all swarm intelligence optimization algorithms, their, social structure and division of labor mechanisms are not different, ABC and SSA represent the social division of labor with a clear and fixed division of labor, and the populations in ABC are divided into three roles (Employed Bees, Onlooker Bees, and Scout Bees), and the populations in SSA are divided into (Producers, Scroungers, Vigilance). In IPO, the population is homogeneous without fixed roles, and in each iteration, any "parrot" may randomly perform any one of the four behaviors (Foraging, Staying, Communicating, Fear), and its behavioral pattern is probabilistic, not Their behavioral patterns are probabilistic rather than determined by their "status" in the population.

Although both IPO and IWOA are group algorithms that simulate animal behavior, their core mathematical models and exploration mechanisms are fundamentally different. Firstly, the core predation model is different, the most unique mechanism in IWOA is the simulation of humpback whale's spiral bubble net predation strategy, which corresponds to a unique logarithmic helix updating formula, and is the core of its localized accurate exploitation, while IPO exploration is mainly achieved by mimicking the optimal individual's "Staying Behavior" and "Communicating Behavior", which are more generalized models; secondly, both of them have a group behavior model, which is more generalized. The exploration process of IPO is mainly realized by mimicking the "Staying Behavior" and "Communicating Behavior" of the optimal individuals, which are more general models of group behavior; secondly, the global exploration mechanisms of IWOA and IPO are different. Secondly, the global exploration mechanisms of IWOA and IPOA are also different. IWOA has a very clear mechanism in its global exploration phase, i.e., the position of an individual is updated not by referring to the optimal solution, but by referring to another randomly selected individual from the population, and this strategy of forcing the individual to deviate from the optimal solution is the key to maintain the diversity of the population and to conduct the global search, while IPOA's global exploration relies on the Levy's flight of large random jumps and the perturbations introduced in the "Fear of Strangers" behavior, which is a different exploration strategy from the IWOA's guidance mechanism based on random individuals [2].

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 4.

Matrix Laboratory (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	34.0450	86.0394	35.7060	33.2665	42.4339
	Average	35.1202	96.5746	38.2419	34.0450	43.7053
convergence time(s)	Slowest	13.85	20.2	15.47	14.04	45.39
	Average	11.21	16.78	14.61	13.29	44.15
	Fastest	10.75	15.6	13.17	12.61	42.39

Figure 4

Simulation environment.

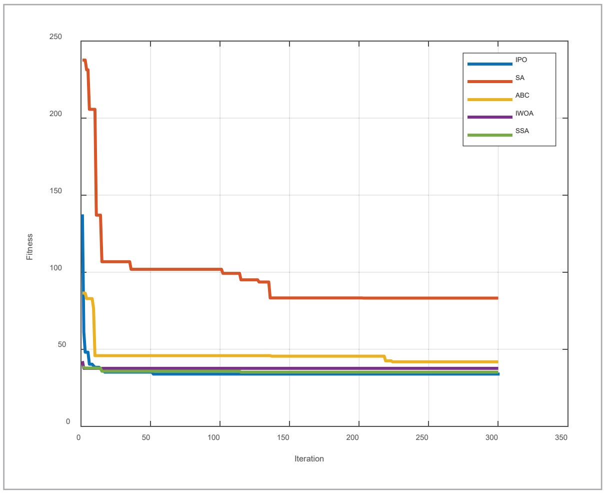


Figure 6

Optimal path for each algorithm.

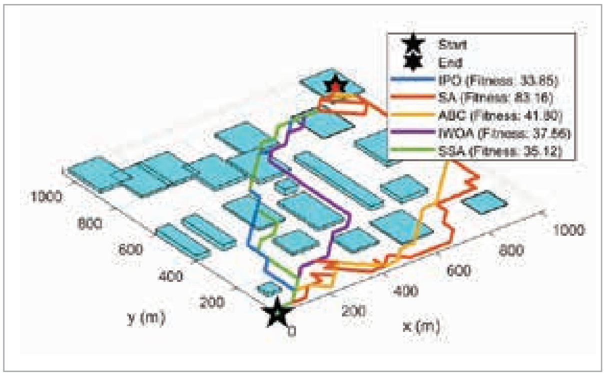
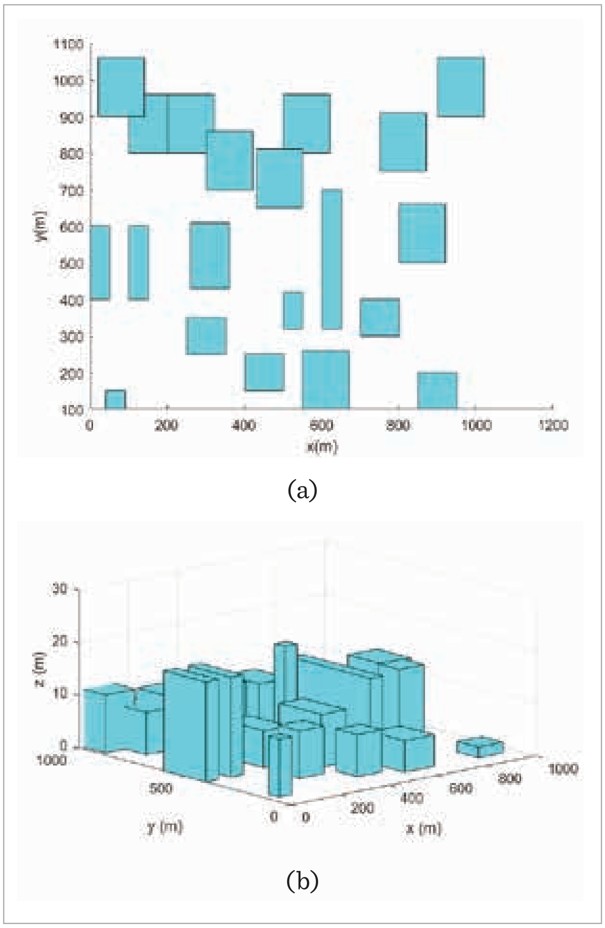


Figure 5

Iteration curve at 300 iterations ((a) for two-dimensional, and (b) for three-dimensional).



(a)

(b)

The iteration curve comparison graph shows the fitness values change during the iteration process for five optimization algorithms (IPO, SA, ABC, IWOA, and SSA). 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 IPO 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. Meanwhile, comparing the optimization curves of each optimization algorithm (as shown in Figure 6), it can be seen that IPO searches in the initial stage with clearer direction and relatively fewer inflection points, and the adaptation value is relatively lower, and the path search ability is better than each optimization algorithm.

In summary, IPO exhibits the best convergence speed and fitness performance in most cases, while

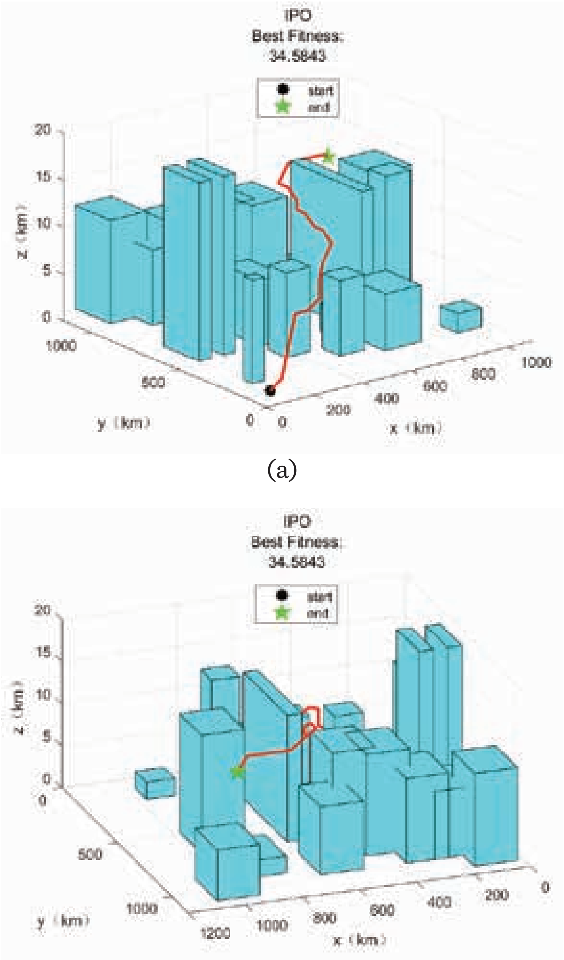
IWOA and SSA perform well but slightly less than IPO, and ABC and SA are relatively weak in convergence ability and effectiveness [12].

Based on this, we use IPO for UAV three-dimensional path planning, and the planned path is shown in Figure 7.

In order to further explore the robustness of the IPO for UAV 3D path planning in more complex environments, we adopt the mountain peak model as an obstacle for UAV flight, and the 3D surface model is used in modeling to make the generated mountain peak environment diverse and realistic, and the map is shown in Figure 8.

Figure 7

Roadmap of the Best Result of the IPO.



(a)

(b)

Figure 8

Mountain Peak Model.

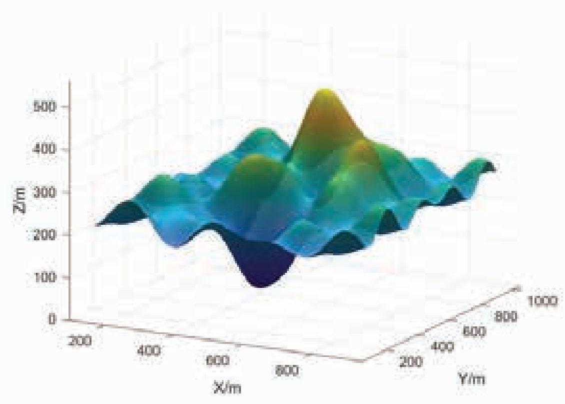


Figure 9

IPO Optimal Path.

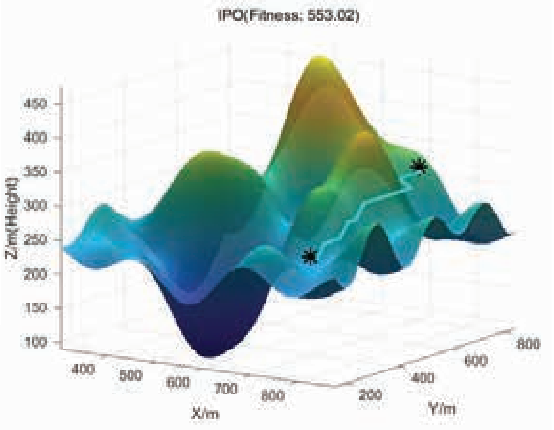


Figure 10

ABC Optimal Path.

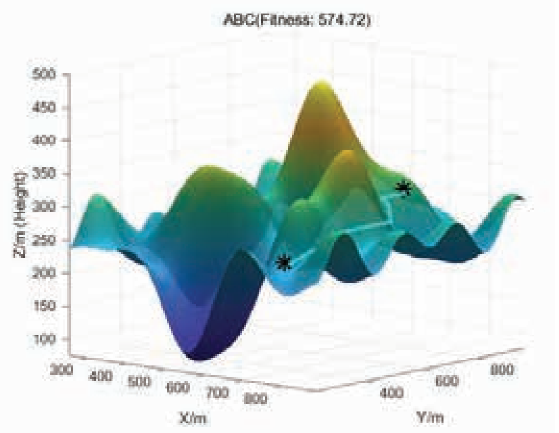


Figure 11

SA Optimal Path.

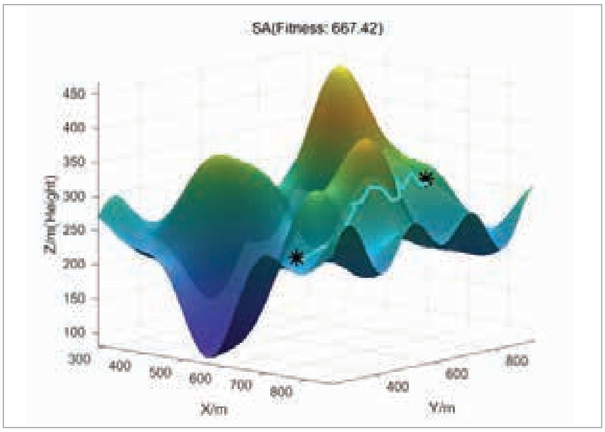


Figure12

WOA Optimal Path.

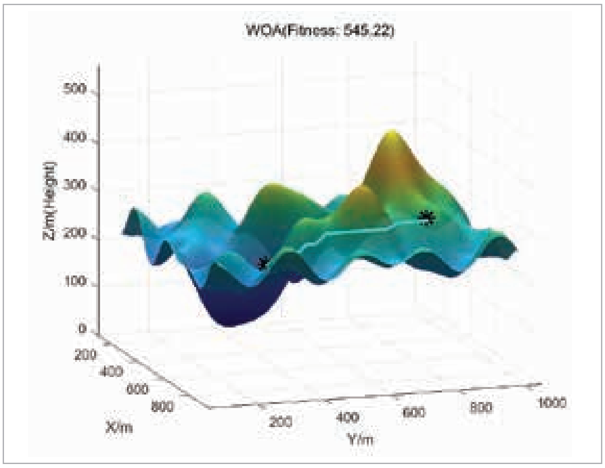
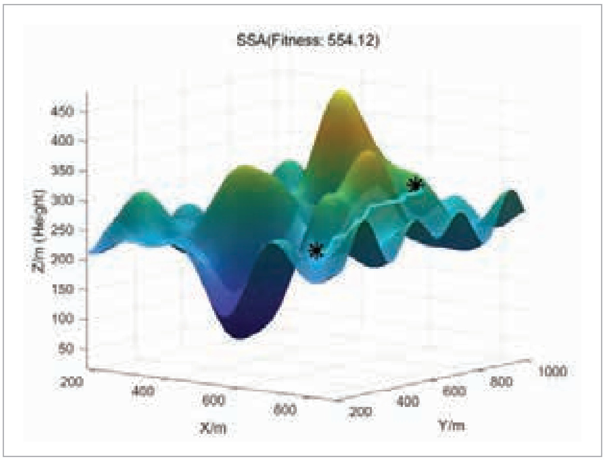


Figure 13

SSA Optimal Path.



In this simulation scenario, the challenge is different from previous scenarios in that there are no discrete obstacles to navigate around, but rather a continuous, undulating costal surface, where the drone needs to find the shortest horizontal projection path while maintaining a safe flight altitude, while minimizing additional energy consumption due to climbing and descending, and avoiding unnecessary "going over the hill". Comparing the planned paths of Figures 9-13, it can be seen that the paths planned by the IPO proposed in this paper are smoother and have fewer inflection points. Its path is very clever, finding a lower "saddle" to cross between the two main peaks, effectively avoiding the need for a major climb. the IPO's dynamic adaptive mechanisms allow it to explore the "col" region early on, while Levy flights Levy Flight and Hybrid Mutation allow IPO to make large jumps, exploring from valley to valley rather than being trapped in a locally optimal "basin", proving once again IPO's ability to strike a balance between global exploration and local exploitation [18].

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 Figure 14 and Tables 2-3 below. Numbers represent mid-points, and letters represent target points.

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.

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

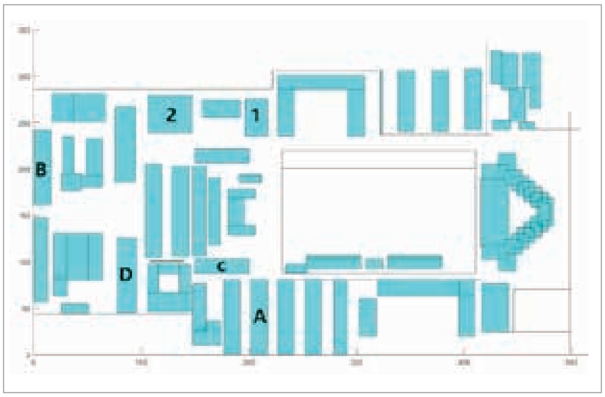
Table 3

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 14

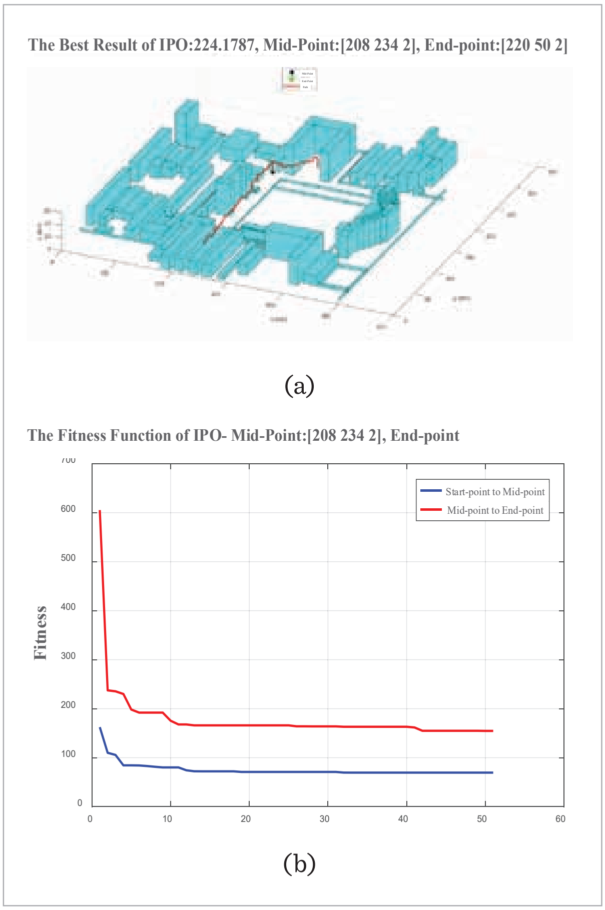
Two-dimensional distribution of coordinates.



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 is 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 Figure 15.

Figure 15

Fengwei Canteen to Residential Building ((a) is the flight path, and (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 can find shorter flight paths, thereby effectively reducing drone flight costs [7].

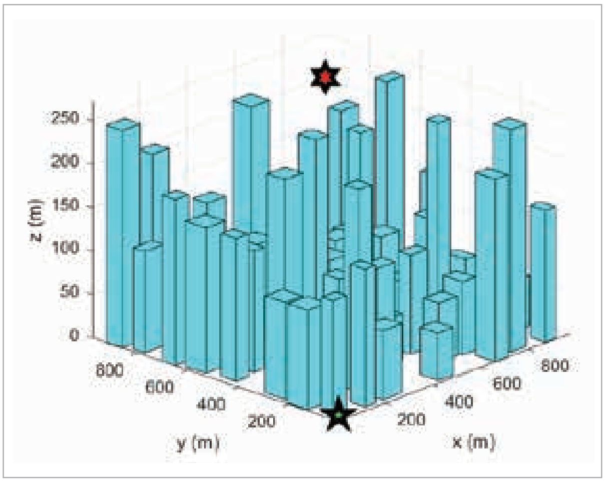
At the same time, the fitness function curve also reveals that the IPO demonstrates rapid convergence in the early stages of iteration, requiring only a few iterations to achieve an optimal solution. It is attributed to the global search advantages of SPM chaotic mapping and the dynamic adjustment capabilities provided by the adaptive switching factor. Additionally, the paths generated by the IPO 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 order to further analyze the effectiveness of IPO in different urban scenarios, we constructed two different scenarios to further demonstrate the effectiveness of the proposed method. Scenario A is a high-density urban core, characterized by tall and dense buildings with small building spacing, forming an “urban canyon”, as shown in Figures 15-16.

Figure 16

High-density urban core.



High-density urban cores mainly test the algorithm's pathfinding ability in the vertical dimension, obstacle avoidance in narrow spaces, and the efficiency of finding the globally optimal path among a large number of potential local optimal solutions (e.g., different paths to bypass a tall building). The fitness curves and optimized routes are shown in Figures 17-18.

Figure 17

Fitness Curve (High-Density Urban).

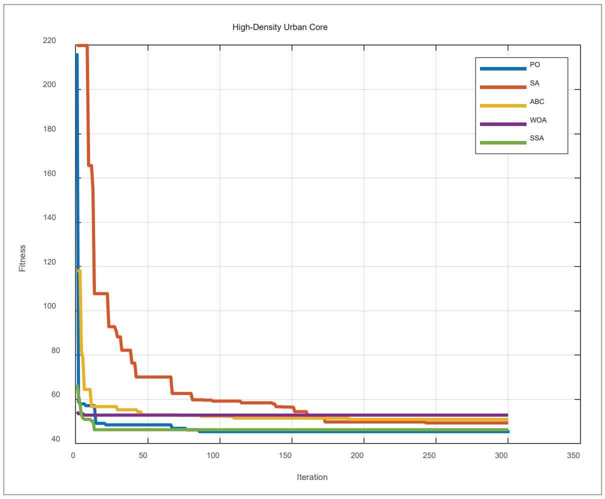
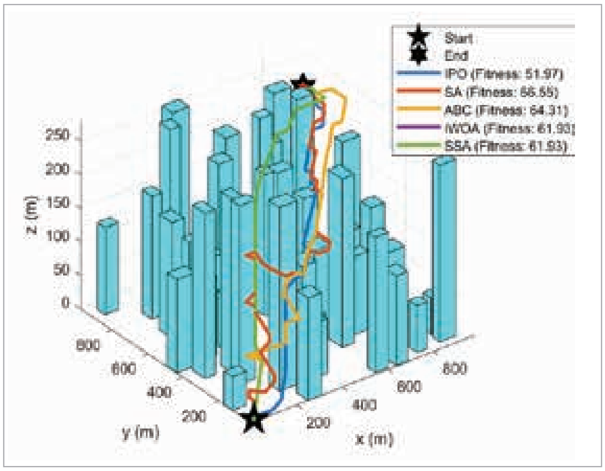


Figure 18

Optimal Path (High-Density Urban).

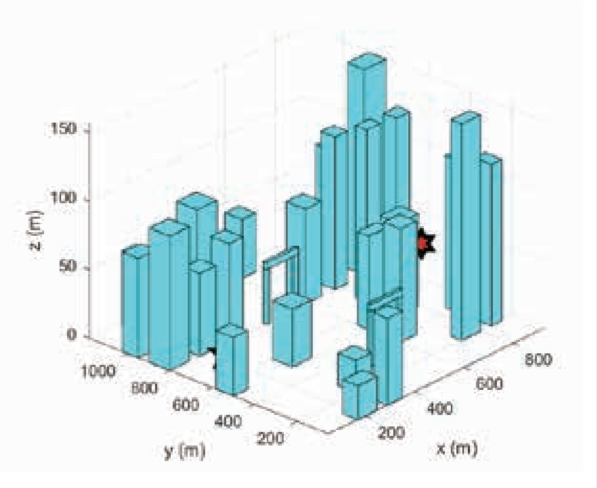


Seeing the convergence curves, IPO, IWOA, and SSA show high performance. They are all swarm intelligent optimization algorithms, which can explore complex urban spaces in parallel using the entire population. IPO performs the most outstandingly. Its dynamic self – adaptive foraging guidance helps it quickly locate high – quality areas of main roads in the early stage, and SPM chaotic mapping and Mixed variants provide population diversity and the ability to jump out of local optimal solutions. SA performs the worst and converges the slowest, indicating that individual search algorithms struggle to explore the whole map efficiently in complex and high - dimensional solution spaces with only random wandering and probabilistic jumps of individual solutions. Meanwhile, observing the optimized route, IPO passes through the densest building center and finds a “shortcut”, proving the effectiveness of its improved strategy in avoiding local optimality. It finds the optimal core path instead of the longer peripheral path.

Subsequently, Scenario B was constructed as an urban area with rivers and bridges, and this scenario introduced multiple levels of access space (over/under bridges) for testing the decision-making ability of IPOs in more complex three-dimensional spaces, as shown in the Figure19.

Figure 19

Urban Riverfront.



This map mainly tests the ability of the algorithms in dealing with irregularly shaped obstacles and multi-level (e.g., over/under bridge) path planning. The fitness curves of each optimization algorithm as well as the Optimal Path are shown in Figures 20-21.

The central challenge of this scenario is a major “strategic choice”: should the drone fly high over the bridges or stay low and pass through the open water between them? This choice divides optimization algorithms into two performance camps.

From Figure 20, IPO, SSA, and IWOA quickly converge to a low fitness interval, meaning they make the correct “macro - decision” of the “low crossing” strategy with a shorter flight distance. As swarm intelligence optimization algorithms, they show high accuracy. ABC converges to a worse plateau, indicating its population chooses the “fly over the bridge” strategy and falls into a local optimum. SA performs worst, reaching a high fitness of nearly 100 early. As an individual search algorithm, it likely makes a wrong start, like entering the river - bank building cluster, and gets trapped in a local optimum due to the high energy barrier, unable to correct its macroscopic route even at high temperatures.

Figure 20

Fitness Curve (Urban Riverfront).

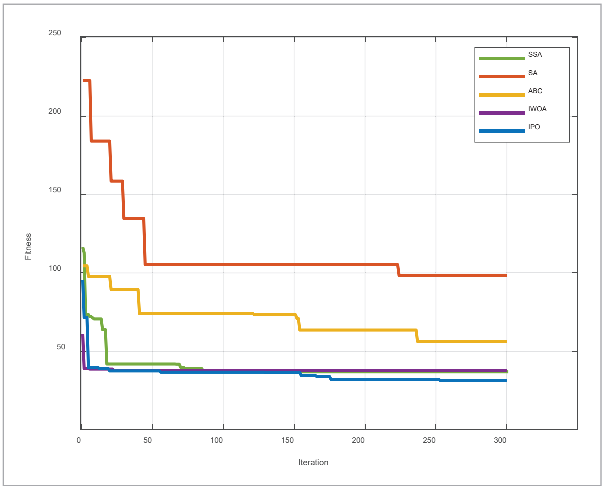
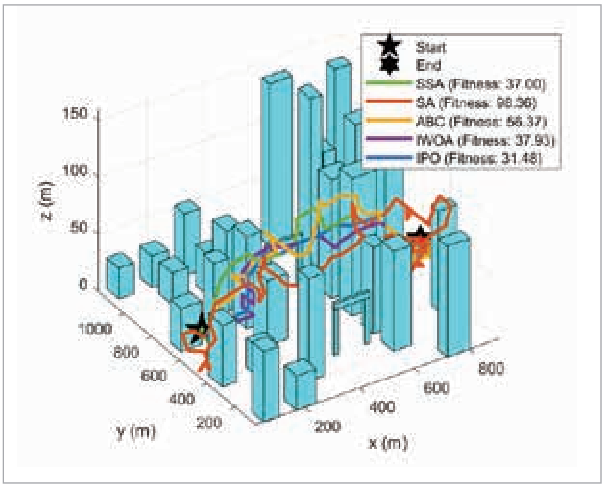


Figure 21

Optimal Path (Urban Riverfront).



The observation of the Optimal Path (Figure 21) visually confirms the convergence curve conclusion. The paths of IPO, SSA, and IWOA show they choose the “low altitude traversal” strategy, staying below the bridge height and flying through the open area.

The IPO's path is the smoothest and most direct, almost a straight line, proving its efficient local path optimization after choosing the correct strategy.

In conclusion, the 3D path planning simulation experiments validate the effectiveness and robustness of the IPO 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, see Figure 23 - Figure 28.

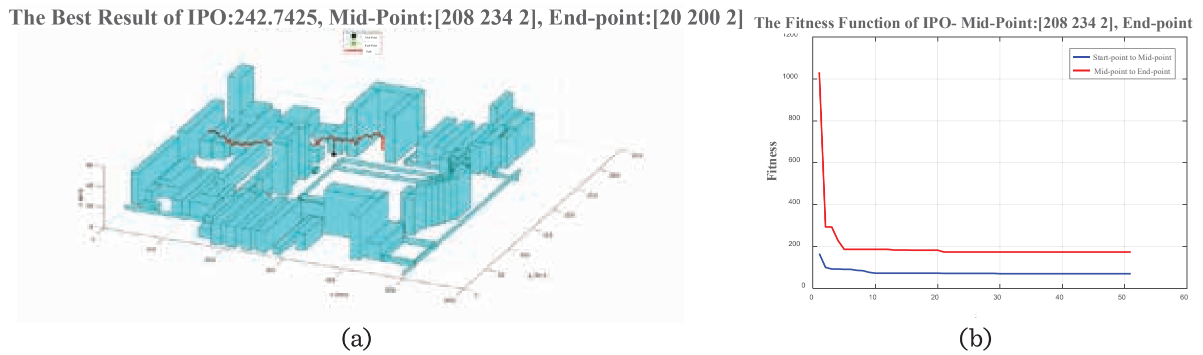
5. Conclusion

This paper addresses the 3D path planning problem of UAVs in complex urban environments and proposes an Improved Parrot Optimizer algorithm. By introducing SPM chaotic mapping, dynamic adaptive foraging mechanisms, adaptive switching factors, and a hybrid Cauchy and Gaussian mutation strategy, the algorithm's global search and convergence speed are enhanced. A 3D path planning solution framework is constructed with a simulation platform, and experiments show the improved algorithm has notable improvements in accuracy, speed, and path smoothness.

The paper also models the BISU campus, High - Density Urban Core, and Urban Riverfront in 3D and conducts path planning simulations, validating the algorithm's potential in real - world environments. However, the research has limitations. Consider exploring the algorithm's application on parallel computing platforms to enhance processing and adaptability to large - scale problems. Apply it to multi - UAV collaborative scenarios for effective coordination. Study the algorithm's adaptability and robust-

Figure 22

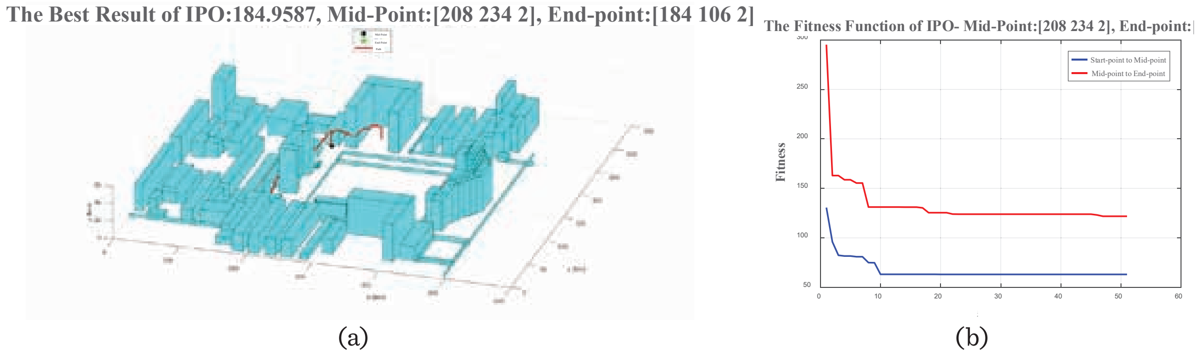
Fengwei Canteen to Building 7 ((a) is the flight path, and (b) is the fitness function curve).



(a) (b)

Figure 23

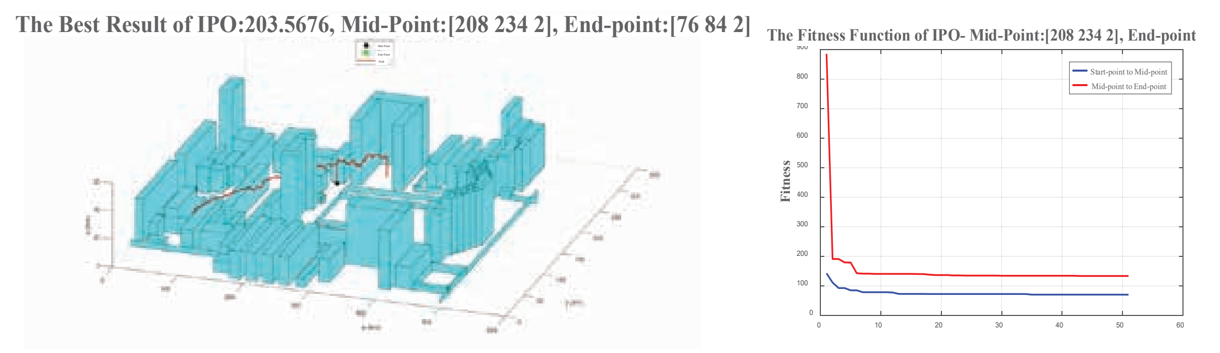
Fengwei Canteen to Building 3 ((a) is the flight path, and (b) is the fitness function curve)



(a) (b)

Figure 24

Fengwei Canteen to Building 1 ((a) is the flight path, and (b) is the fitness function curve).

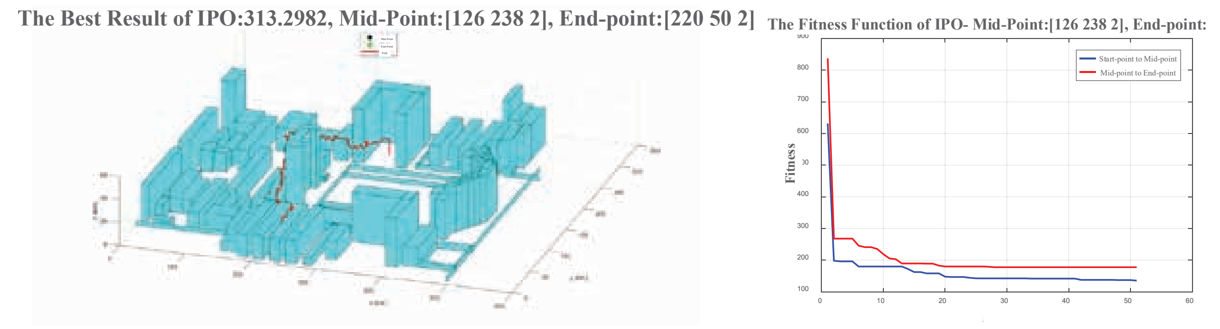


(a)

(b)

Figure 25

Xiangyu Canteen to Residential Building ((a) is the flight path, and (b) is the fitness function curve).

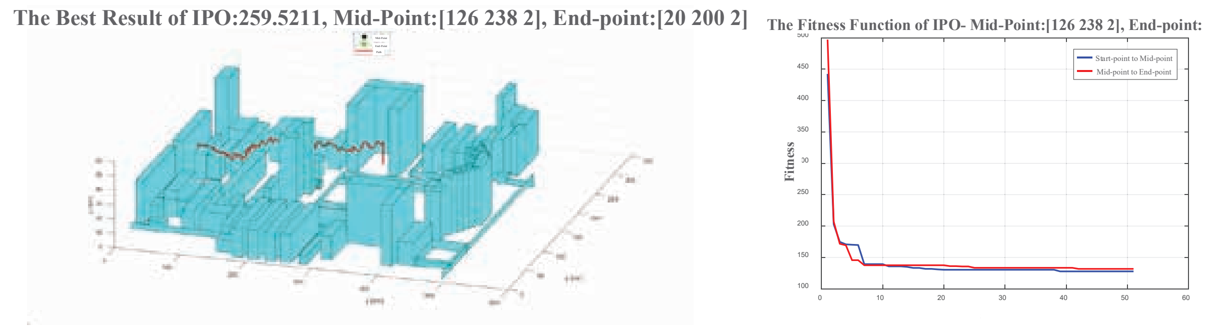


(a)

(b)

Figure 26

Xiangyu Canteen to Building 7 ((a) is the flight path, and (b) is the fitness function curve).

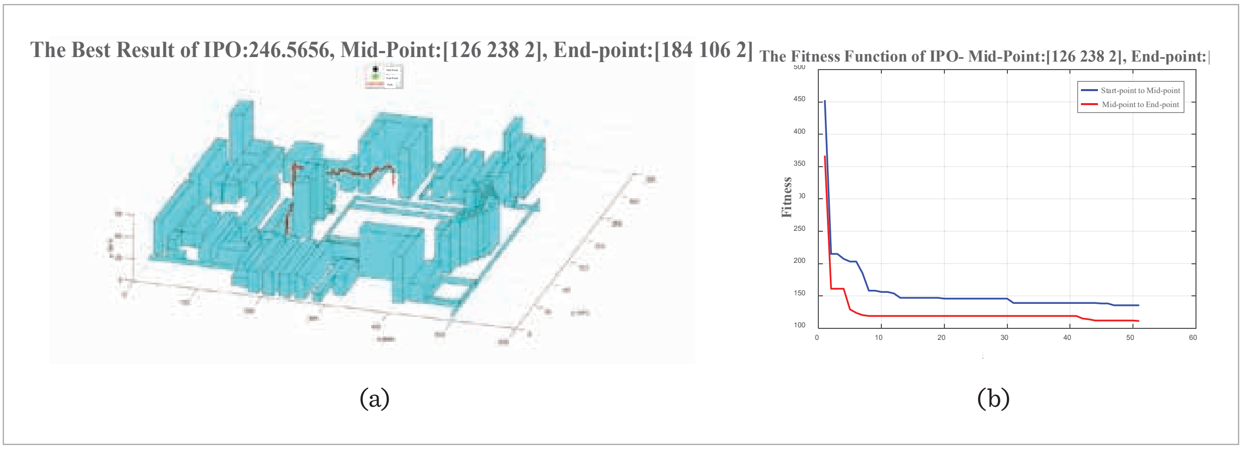


(a)

(b)

Figure 27

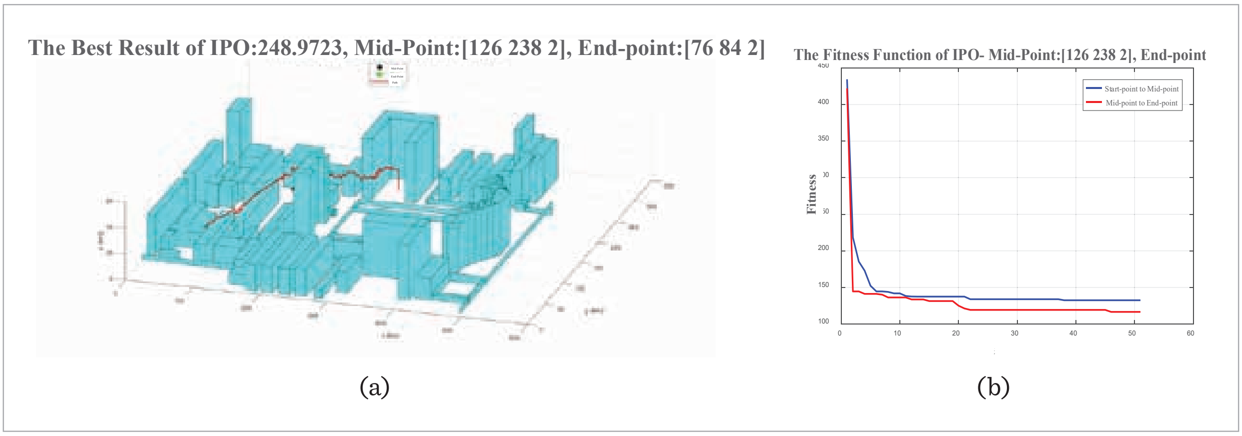
Xiangyu Canteen to Building 3 ((a) is the flight path, and (b) is the fitness function curve).



(a) (b)

Figure 28

Xiangyu Canteen → Building 1 ((a) is the flight path, and (b) is the fitness function curve).



(a) (b)

ness under dynamic constraints like wind speed. Employ curve smoothing strategies and develop adaptive mechanisms for automatic parameter adjustment. Conduct actual UAV flight tests outside the lab.

The proposed improved IPO algorithm offers an effective solution for UAV 3D path planning with good engineering application prospects. Future research will further optimize and apply the algorithm to promote UAV technology development in more fields.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Appendix A

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

References

1. Alejo, D., Cobano, J. A., Heredia, G., Ollero, A. Optimal Reciprocal Collision Avoidance with Mobile and Static Obstacles for Multi-UAV Systems. 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014, 1259-1266. https://doi.org/10.1109/ICUAS.2014.6842383
2. Ao, T., Luo, Y., Ye, J., Wang, Q., Chi, Y., Zhou, N. Joint Trajectory and Energy Optimization for Wirelessly Charged UAVs in Power Transmission Line Inspections. Wireless Networks, 2025, 31(5), 3893-3906. https://doi.org/10.1007/s11276-025-03975-7
3. Arslan, O., Berntorp, K., Tsiotras, P. Sampling-Based Algorithms for Optimal Motion Planning Using Closed-Loop Prediction. 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, 4991-4996. https://doi.org/10.1109/ICRA.2017.7989581
4. Bui, D. N., Duong, T. N., Phung, M. D. Ant Colony Optimization for Cooperative Inspection Path Planning Using Multiple Unmanned Aerial Vehicles. 2024 IEEE/SICE International Symposium on System Integration (SII), 2024, 675-680. https://doi.org/10.1109/SII58957.2024.10417512
5. Cabreira, T. M., Brisolara, L. B., Ferreira Jr., P. R. Survey on Coverage Path Planning with Unmanned Aerial Vehicles. Drones, 2019, 3(1). https://doi.org/10.3390/drones3010004
6. Fu, Y., Ding, M., Zhou, C. Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAV. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 2012, 42(2), 511-526. https://doi.org/10.1109/TSMCA.2011.2159586
7. Hui, F., et al. A Review of Research and Applications in Low-Altitude Logistics and Delivery Using Unmanned Aerial Vehicles. Industrial Engineering Journal, 2025, 28(1), 9-21. https://doi.org/10.3969/j.issn.1007-7375.240388.
8. Iacono, M., Sgorbissa, A. Path Following and Obstacle Avoidance for an Autonomous UAV Using a Depth Camera. Robotics and Autonomous Systems, 2018, 106, 38-46. https://doi.org/10.1016/j.robot.2018.04.005
9. Li, Y., Liu, M., Jiang, D. Application of Unmanned Aerial Vehicles in Logistics: A Literature Review. Sustainability, 2022, 14(21). https://doi.org/10.3390/su142114473
10. Lin, S., Li, F., Li, X., Jia, K., Zhang, X. Improved Artificial Bee Colony Algorithm Based on Multi-Strategy Synthesis for UAV Path Planning. IEEE Access, 2022, 10, 119269-119282. https://doi.org/10.1109/ACCESS.2022.3218685
11. Lyridis, D. V. An Improved Ant Colony Optimization Algorithm for Unmanned Surface Vehicle Local Path Planning with Multi-Modality Constraints. Ocean Engineering, 2021, 241, 109890. https://doi.org/10.1016/j.oceaneng.2021.109890
12. Mohd Daud, S. M. S., et al. Applications of Drone in Disaster Management: A Scoping Review. Science & Justice, 2022, 62(1), 30-42. https://doi.org/10.1016/j.scijus.2021.11.002
13. Phung, M. D., Quach, C. H., Dinh, T. H., Ha, Q. Enhanced Discrete Particle Swarm Optimization Path Planning for UAV Vision-Based Surface Inspection. Automation in Construction, 2017, 81, 25-33. https://doi.org/10.1016/j.autcon.2017.04.013
14. Qu, C., Gai, W., Zhang, J., Zhong, M. A Novel Hybrid Grey Wolf Optimizer Algorithm for Unmanned Aerial Vehicle (UAV) Path Planning. Knowledge-Based Systems, 2020, 194, 105530. https://doi.org/10.1016/j.knosys.2020.105530
15. Roberge, V., Tarbouchi, M., Labonte, G. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. IEEE Transactions on Industrial Informatics, 2013, 9(1), 132-141. https://doi.org/10.1109/TII.2012.2198665
16. Shan, Z., Wang, Y., Liu, X., Wei, C. Fuzzy Automatic Disturbance Rejection Control of Quadrotor UAV Based on Improved Whale Optimization Algorithm. IEEE Access, 2023, 11, 69117-69130. https://doi.org/10.1109/ACCESS.2023.3292265
17. Shao, Z., Yan, F., Zhou, Z., Zhu, X. Path Planning for Multi-UAV Formation Rendezvous Based on Distributed Cooperative Particle Swarm Optimization. Applied Sciences, 2019, 9(13). https://doi.org/10.3390/app9132621
18. Wang, H., Lyu, W., Yao, P., Liang, X., Liu, C. Three-Dimensional Path Planning for Unmanned Aerial Vehicle Based on Interfered Fluid Dynamical System. Chinese Journal of Aeronautics, 2015, 28(1), 229-239. https://doi.org/10.1016/j.cja.2014.12.031
19. Yan, M., Yuan, H., Xu, J., Yu, Y., Jin, L. Task Allocation and Route Planning of Multiple UAVs in a Marine En-

vironment Based on an Improved Particle Swarm Optimization Algorithm. EURASIP Journal on Advances in Signal Processing, 2021, 2021(1), 94. https://doi.org/10.1186/s13634-021-00804-9

20. Yu, X., Jiang, N., Wang, X., Li, M. A Hybrid Algorithm Based on Grey Wolf Optimizer and Differential Evolution for UAV Path Planning. Expert Systems with Applications, 2023, 215, 119327. https://doi.org/10.1016/j.eswa.2022.119327
21. Zhang, J., Zhu, X., Li, J. Intelligent Path Planning with an Improved Sparrow Search Algorithm for Workshop UAV Inspection. Sensors, 2024, 24(4), 1104. https://doi.org/10.3390/s24041104
22. Zhang, Y., Wang, S., Ji, G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. Mathematical Problems in Engineering, 2015, 2015(1), 931256. https://doi.org/10.1155/2015/931256
23. Zhao, Y., Zheng, Z., Liu, Y. Survey on Computational-Intelligence-Based UAV Path Planning. Knowledge-Based Systems, 2018, 158, 54-64. https://doi.org/10.1016/j.knosys.2018.05.033
24. Zhu, W., Duan, H. Chaotic Predator-Prey Biogeography-Based Optimization Approach for UCAV Path Planning. Aerospace Science and Technology, 2014, 32(1), 153-161. https://doi.org/10.1016/j.ast.2013.11.003