

Particle Swarm Optimization Bat Algorithm Path Automatically Planning Research for Police Drones in Hilly Cities

Jing XUE

Chongqing Three Gorges University, Chongqing 404020, China
E-mail: jingxue688@gmail.com

Zefu TAN

Chongqing Three Gorges University, Chongqing 404020, China

Nina DAI

Chongqing Three Gorges University, Chongqing 404020, China

Guoping LEI

Chongqing Three Gorges University, Chongqing 404020, China

Chao HE

Chongqing Three Gorges University, Chongqing 404020, China
E-mail: hechao@sanxiau.edu.cn

Abstract Mountain cities are complex asymmetric dynamic network architectures, and the flight of UAVs in this environment is subject to various constraints, while efficiency is a crucial factor in the trajectory planning of police UAVs, which need to maintain high efficiency and safe flight paths between their starting and ending points, but the traditional trajectory planning method cannot meet the requirements of rapid maneuvering of police UAVs. To achieve this, a 3D terrain map is built, an objective function is established for the flight cost in the UAV trajectory planning process, and a planning algorithm called particle swarm optimization bat algorithm (PSOBA) is proposed. PSOBA combines the characteristics of the bat algorithm (BA) and the particle swarm optimization algorithm (PSO) to improve population diversity and resolve the delayed convergence issue in the last phases of BA. Simulation results show that PSOBA is more effective than BA, with a search time for the best solution that is approximately 20.43% shorter and a convergence value of the objective function that is approximately 38% smaller. PSOBA is also able to plan a quicker, shorter, and safer flight path compared to other trail planning algorithms that enhance the bat algorithm. These findings suggest that PSOBA is a powerful algorithm with potential application value in UAV trajectory planning control in the mobile intelligence era. Contribute to the service of public social security.

Keywords hilly cities; police drones; asymmetrical; trajectory planning; particle swarm bat algorithm

Received April 5, 2023, accepted October 7, 2023

Supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (KJZD-K202201203); Chongqing Three Gorges University and the Public Security Bureau of Wanzhou District, Chongqing (wzstc20210302, wzstc20220310)

1 Introduction

Unmanned aerial vehicles (UAVs) are aircraft that do not require a pilot on board and are controlled by radio signals and automated programming. They are commonly used for complex missions due to their quiet operation, flexibility, and advanced technology integration^[1–4]. Planning the route for the UAVs has become increasingly important for successful mission execution, as it lays the groundwork for autonomous navigation and mission planning^[5–7].

Efficient and precise trajectory planning is crucial for improving the responsiveness of police UAV broadcasts in unpredictable situations^[8–12]. In mountainous cities, traffic accidents can be more challenging to manage due to the hilly terrain and the risk of parked cars sliding backward, posing a severe safety threat. In such situations, it may be difficult for police officers to access the accident site promptly. To overcome this challenge, emergency drones can be utilized, which can efficiently plan the best flight path to reach the site quickly. This allows the drone to gather and transmit essential site data to the command center, which can then develop an effective action plan based on the situation on the ground. By using drones, police officers can reduce their workload, ease traffic congestion and ensure public safety^[13–15]. Hence, drones must plan a safe, efficient, and seamless flight path to carry out their tasks effectively.

Airspace modeling and trajectory search operations are both important parts of the difficult process of UAV trajectory planning in mountainous cities. Initially, airspace modeling integrates flight space into distinct abstract areas, which serves as the foundation for trajectory planning. For police UAVs, each mission is closely related to the safety of people's lives, so it is necessary to make a more demanding choice of the flight path^[16]. In comparison to plain area airspace modeling, mountainous urban airspace modeling can be much more complex due to the increased number of obstacles and decreased fault tolerance in the flight domain. The high information density in the physical space of mountain towns must be taken into account for accurate and reliable mountain city airspace modeling^[17, 18]. Hence, great resolution of the abstract space must be guaranteed by mountain city modeling techniques.

The practical applicability of traditional airspace modeling techniques is constrained, particularly for complicated barrier boundaries such as irregular surfaces that could introduce modeling bias. In response, Chen, et al. constructed Voronoi diagrams based on each threat center and used an artificial potential field approach to generate local tracks. For tracking points within the Voronoi diagram, the moving target is tracked along one side of the Voronoi diagram using Dijkstra's algorithm^[19]. However, the construction of Voronoi diagrams can be difficult when the null domain is too complex. Rivera et al combined Manhattan distance and Octel distance to define a 2^k recursive neighborhood and a 2^k distance function in a two-dimensional setting, which can be used as a heuristic, and proposed a 2^k neighborhood implementation of jump-point search, which has a short running time and also opens the door for future research on the arbitrary angle of incremental search opens the door to future research^[20]. However, the application of this method to a 3D environment assumes that the flight altitude of the UAV remains constant, making it difficult to generalize the method. The size of the neighborhood also affects the efficiency of the UAV's trajectory planning.

The creation of planning algorithms is a key component of trajectory search. Trajectory planning in complex mountainous urban areas must guarantee that each step of the UAV's flight

in the 3D flying airspace is precise and secure. Three different types of UAV track planning algorithms are currently in widespread use: conventional track planning algorithms, heuristic track planning algorithms, and swarm intelligence bionic track planning algorithms. To solve the issue of classic trajectory optimization techniques' lengthy processing times, Wu, et al. combine the bi-directional artificial potential field method with bi-directional bias sampling, which cleverly varies the sampling space to nicely reduce sampling of irrelevant space and improve the convergence ratio^[21]. Huang, et al. suggested a bi-directional fast exploration random tree algorithm based on the greedy strategy optimization algorithms' productivity and stability in varied situations can be improved. The expansion of the algorithm nodes is also improved to enhance the solving efficiency^[22]. However, the above method takes a long time to compute and is not effective in practice.

To address the drawback that the heuristic algorithm has low search efficiency and can only generate one path when the operation area is large, Bai, et al. improved the knowledge about the flight environment and the motion limitations of the typical A* algorithm UAV and adaptively adjusted the evaluation function based on the safety threshold, obstacles and environmental information of the UAV to alleviate the limitations of the DWA algorithm in terms of local optimization and planning time^[23]. Sichen, et al. employed a simulated annealing approach to compute approximate optimized pathways inside a sub-region, in order to reduce a single UAV's energy loss and to establish a balanced distribution of duties within a UAV formation. Compared to a traditional zigzag path, the path length is cut by about 12.6%^[24]. However, such algorithms require powerful computing power, which is not conducive to police UAVs performing fast-response, mobile, and flexible combat missions.

To address the shortcomings of the swarm intelligent bionic trajectory planning algorithm in relation to local optimality, Huang, et al. recommended a better method based on spherical vectors, this took into account the correlation between the search space and the UAV's inherent constraints^[25]. The improved algorithm has stable convergence and improves the safety in addition to the viability of the selected paths. Zhang, et al. offered a new by using a genetic algorithm mixed initialization methods, firstly, in order to avoid redundant paths and inhibit the algorithm from entering a near - optimal solution, the duplicative clusters or redundant paths in the path are decided to delete. Three free meshes are randomly chosen as the central nodes, after which one part uses the random initialization method and the other part uses the greedy correlation technique to generate the direction^[26]. Wang, et al. made the initial suggestion for improving the BA method by including differential evolution (DE) to optimize the 3D path planning problem. In IBA, the variation operator of DE was used for bat selection during the position update process. By connecting the selected nodes using IBA, a safe path was successfully obtained. The experimental results show that IBA accomplishes the 3D path planning problem better than the basic BA model^[27]. Lin, et al. used a fusion algorithm combining artificial potential field and chaotic bat algorithm so that the initial distribution of bats can be randomized and homogenized, thus the solution space can be traversed, on the basis of this, to avoid the algorithm reaching a local optimum, they recommended an adaptive parameter updating technique. The approach's success in resolving multi-objective optimization problems was shown by the simulation results^[28]. These algorithms' convergence speed and

stability have significantly increased over earlier trajectory planning techniques. It has been demonstrated, nonetheless, that the majority of swarm intelligent bionic trajectory algorithms for UAV route planning problems still suffer from performance issues, particularly regarding solution accuracy, smooth path trajectories, and sporadically falling into local optimum.

Based on the preceding research, there is no optimal solution for UAV trajectory planning in complicated situations that can concurrently handle the following problems: 1) 3D modeling based on complicated settings in hilly cities; 2) rapid and efficient flight pathways; 3) smooth flight trajectories.

The following techniques are used in this research to resolve the trajectory planning issue for police UAVs in the aforementioned complex environment.

I: Using a three-dimensional grid method, the cost function is set to reflect the various threats faced by the UAV and the various constraints on the flight path. The three-dimensional space is broken down into several horizontal planes perpendicular to the Z -axis, and each horizontal plane is then divided into a grid to reduce the computational complexity. Generally speaking, the trajectory is better and adoption is easier the narrower the cost function.

II: B-sample curve method is introduced for pre-processing to enhance the fit to the flight trajectory and to enhance the robustness of the flight trajectory.

III: It is suggested that the particle swarm algorithm be grafted into the bat algorithm. This would improve the sluggish convergence speed, low convergence accuracy, and propensity to fall into local extremes in the late stages of the bat algorithm, as well as create a safe and effective flight path.

The remainder of the essay is structured as follows. The cost function setup, smoothing technique, and modeling of the UAV flight trajectory planning problem are all covered in the second section. The final section goes into detail about the particle swarm bat algorithm (PSOBA). The analytical and experimental results are presented in Part IV. In Part V, we put an end to this research effort and offer a look ahead.

2 Model Construction

2.1 Environmental Modeling

The modeling of mountainous cities has to take into account the special characteristics of mountainous cities compared to plain areas. In addition to the standing tall buildings, the terrain places higher demands on the planning of flight paths, and it is also crucial to find a suitable environment modeling method in conjunction with the UAVs applied in this paper in specific areas^[29-32].

In the coordinate system XOY, the drone's beginning is indicated by the following locations: S and the ending point as E . There are obstacles between S and E . A safe and efficient flight path needs to be planned between S and E in a reasonable way.

First, connect points S and E , then divide the SE segment into $P + 1$ equal parts. At each segmented point, make a vertical line of the SE segment, noted as $k_1, k_2, \dots, k_j, \dots, k_P$, take a discrete point at each vertical segment k_j and connect these discrete points in sequence to form the flight path of the UAV, as shown in Figure 1. Hence, the trajectory planning issue is changed into a multi-objective constraint problem that is optimized to determine the cost

function's ideal adaption value.

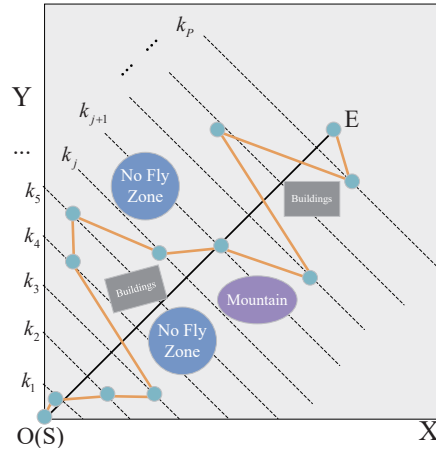


Figure 1 Simulation of flight path in a two-dimensional environment

While modeling the environment in 3D space, a 3D grid approach is used to establish a 3D right-angle coordinate system OXYZ. The region OABC-MNPQ in the coordinate system is selected, where the plane OABC is on a plane OXY. Next, OABC-MNPQ is divided into m parts along the X -axis, OABC-MNPQ is divided into n parts along the Y -axis, and OABC-MNPQ is divided into l parts along the Z -axis. The planning space will be divided into $m \times n \times l$ cubes, as shown in Figure 2.

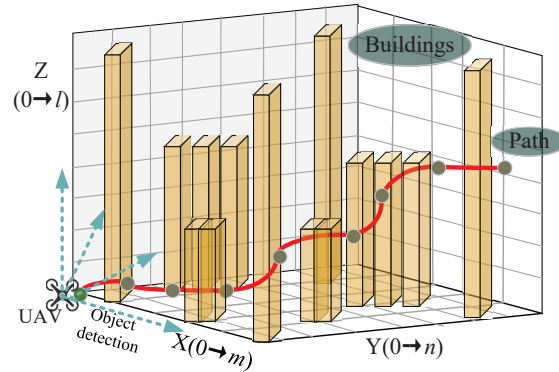


Figure 2 Three-dimensional spatial modeling grid

For different-sized UAVs and professional settings, the grid of units grid lengths on the XOY plane and Z -axis can be reset according to the UAV's own obstacle-crossing capability, obstacle size, mountain height, and hillside angle.

After the aforementioned processing, a collection of nodes serves as the UAV's working area, and the following equation can be used to translate the length of the UAV flight track into the combination of the unique points' distances^[33].

$$D = \sum_{i=1}^{p-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}, \quad (1)$$

where p is the total number of discrete points flown by the UAV.

2.2 Cost Assessment Models

The UAV flight track cost assessment must take into account three elements: energy consumption, flight height, and collision threat^[34]. The fuel consumption cost is proportional to the total flight range, the altitude cost is determined by the altitude at which the UAV flies, and the collision threat mainly considers the probability of collision between the UAV and potential hazards during flight.

The cost of fuel consumption is defined, as shown in Equation (2).

$$f_1 = W \sum_{i=1}^n l_{ij} + \sum_{i=1}^n E_0 \frac{h}{H} dh, \quad (2)$$

where W represents the fuel consumption per unit length of the UAV during the mission, H define the safe altitude at which the UAV must not exceed, E_0 represents the energy cost of maintaining the UAV at a certain altitude, and h define the current altitude of the UAV.

The flight level threat is defined, as shown in Equation (3).

$$f_2 = \sum_{i=1}^n g_i, \quad (3)$$

$$g_i = \begin{cases} C_1(H_{\min} - h_i), & h_i < H_{\min}, \\ 0, & H_{\min} < h_i < H_{\max}, \\ C_1(h_i - H_{\max}), & h_i > H_{\max}, \end{cases}$$

where g_i represents the severity of the threat if the drone is flying too high, h_i describe the average flight altitude in the paragraph i , $H_{\max}$ the maximum altitude at which the drone can fly, $H_{\min}$ specify the drone's distance from the ground at the lowest safe altitude, c_1, c_2 define normal number. The collision threat is defined, as shown in Equation (4).

$$f_3 = \int_0^{L_{ij}} \sum_{k=1}^{N_t} \frac{t_k}{[(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2]^2} dl, \quad (4)$$

where t_k describe the threat factor, which measures the threat level between the threat source and the drone node. N_t represents the total number of threat sources, the coordinates of the drone (x, y, z) , and the coordinates of the center of the threat source (x_k, y_k, z_k) .

2.3 UAV Manoeuvrability Constraint Model

Finding the best flight path from the mission's departure position to the destination point is the main objective of the UAV trajectory planning problem. It is vital to take into account the UAV's own limits in addition to the environmental information of the mission space, which mainly include turn radius constraints, pitch angle constraints and minimum inertia distance constraints^[35].

The UAV turning radius is the turning arc of the UAV from one sub-path segment to the next during flight. In an effort to ensure the UAV's air safety, a maximum steering angle $\psi_{\max}$ needs to be set during UAV trajectory planning. Assuming that the horizontal projection of

the i path segment is $a_i = (x_{i+1} - x_i, y_{i+1} - y_i)^T$, Equation (5) should be satisfied between adjacent path segments.

$$\frac{a_i^T a_{i+1}}{\|a_i\| \|a_{i+1}\|} \geq \sin \psi_{\max}. \quad (5)$$

Pitch angle constraint means that when the UAV climbs or dives, it will be subject to certain restrictions, mainly in the form of the maximum angle size between the axis of the fuselage and the horizontal plane in the vertical direction cannot exceed the maximum pitch angle that the UAV itself can withstand, then the maximum pitch angle constraint should satisfy Equation (6).

$$\frac{|z_{i+1} - z_i|}{|\vec{a}_i|} \leq \tan \theta_{\max}, \quad (6)$$

where $\theta_{\max}$ describe the maximum pitch angle and z_i, z_{i+1} are the Z -axis coordinates of the start and end points of the i section of the path.

The minimum inertia distance is the shortest distance the UAV needs to travel in its original direction when suddenly changing its current direction. This study searches the nearby grids divided in the built environment planning model for the UAV trajectory, so the grid size can be set to the size of the minimum inertia distance. Let the UAV have n segments of trajectory $\{l_i | i = 1, 2, \dots, n\}$ and let the minimum UAV trajectory segment length be $l_{\min}$, then the minimum inertia distance constraint is shown in Equation (7).

$$l_i \geq l_{\min}, \quad i = 1, 2, \dots, n. \quad (7)$$

In this regard, the following function is chosen as the UAV flight track cost assessment model in this paper.

$$f_{\min} = w_1 f_1 + w_2 f_2 + w_3 f_3 + (1 - w_1 - w_2 - w_3) f_4, \quad (8)$$

where w_1, w_2, w_3 define a weighted value, f_1 describe the fuel cost, which is proportional to the flight range if the UAV maintains a constant speed in flight, f_2 is the altitude threat, f_3 represents the collision threat, and f_4 represents the UAV maneuverability constraint threat.

2.4 Pre-Processing of Track Smoothing Strategies

After the trajectory planning, the generated flight path consists of a series of straight-line segments. This planning result has a guiding effect and does not meet the actual flight characteristics of smooth UAV turns, and in mountainous cities, the probability of collision of UAVs due to excessive turn angles is greater compared to flying in plain areas, which is not practical^[36, 37]. Therefore, after generating the optimal path in the discretized grid, the resulting folded flight trajectory needs to be smoothed. A feasible smoothing result needs to satisfy two requirements, the first being that the curve is smooth everywhere and the second being that the curve cannot create intersections with threatening objects in the environment. Considering that *Bézier Curves* cannot locally modify the UAV flight trajectory, this paper introduces the *B* spline curve method for trajectory smoothing, because *B* spline curves allow local control surfaces and curves, their polynomials' number may not be determined by the amount of command posts^[38, 39]. The K times *B* spline curve expression is Equation (9).

$$P(t) = \sum_{i=0}^n B_{i,k}(t) P_i, \quad (9)$$

where $B_{i,k}(t)$ define the K times B spline basis function, which satisfies the following recursive equation, which is shown in Equation (10).

$$K = 0, B_{i,0}(t) = \begin{cases} 1, & t \in [t_i, t_{i+1}], \\ 0, & \text{Otherwise,} \end{cases} \quad (10)$$

$$K > 0, B_{i,K}(t) = \frac{t - t_i}{t_{i+K} - t_i} B_{i,K-1}(t) + \frac{t_{i+K+1} - t}{t_{i+K+1} - t_{i+1}} B_{i+1,K-1}(t).$$

A smooth flight path is obtained after this method. As can be seen in Figure 3, the adjusted flight path is more feasible.

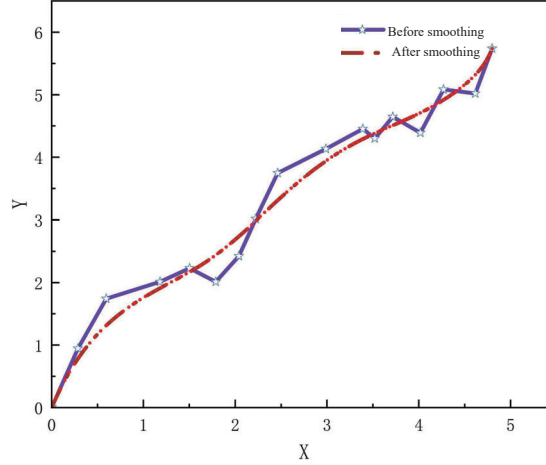


Figure 3 Flight trajectory after B-spline method processing

3 Particle Swarm Bat Fusion Algorithm

3.1 Introduction to Particle Swarm Algorithms

The particle swarm algorithm (PSO) is bionic intelligent optimization algorithm that is inspired by the cooperative foraging behavior of birds. It operates on the following principle: Each solution to an optimization problem is viewed as a bird, or a particle, and each particle moves in a multi-dimensional search space to search for the optimal solution, with each movement representing one iteration. To find the best answer, each particle moves in a multidimensional search space. The fitness value computed by the fitness function is used to determine each particle's current position, and the particle relies on its own "memory" function to store both its ideal position and the population's optimal position. Each iteration represents one iteration. Each movement of a particle is determined by its velocity vector, which is influenced by three factors: The "inertial component", "self-awareness" and "population experience"^[40-43]. The PSO algorithm begins by creating a population of particles in the viable solution space, each of which represents a potential optimal solution to the extreme value optimization issue. The particles are then given positions, velocities, and fitness values to describe them. The algorithm is represented by Equations (11) and (12).

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_g^t - x_{id}^t), \quad (11)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}. \quad (12)$$

In Equation (11), t is the particle's current number of iterations, v_i^t is the velocity of the i particle in the t third iteration; p_i^t is the best position of the i particle in the t iteration, p_g^t is the best position of all particles from the first iteration to the t iteration; x_i^t is the i particle in the t iteration; ω represents the inertia weight, which is mainly used to measure the ratio of global search and local search, when the value of ω is small, do a small local search, otherwise do a global search process, according to practical tests, the value of ω is generally set to 0.729; c_1, c_2 is a factor of speed control, can use the current historical optimal value of individual particles and the global optimal value of the group of particles to fully regulate the search rate, and $0 \leq c_1 c_2 \leq 2$, usually take the value of $c_1 = c_2 = 1.5$, $r_1 r_2$ is a uniformly distributed random number, and $0 \leq r_1 r_2 \leq 1$; in Equation (11) as well as Equation (12), d denotes the particle. The variable v_{id}^{t+1} represents the velocity of the i particle in the d -dimensional space after the $(t+1)$ iteration; accordingly, the variable x_{id}^{t+1} represents the position of the i particle in the d -dimensional space after the $(t+1)$ iteration.

3.2 Introduction to the Bat Algorithm

The bat algorithm was inspired by the echolocation of miniature bats and was proposed by Xin-She Yang in 2010^[44–46]. Most miniature bats radiate sound into their surroundings, identifying prey and avoiding obstacles by picking up echoes from different objects. The loudness is usually highest at the beginning of the prey search, which facilitates the search of a larger space, and then gradually decreases while the pulse frequency gradually increases, which facilitates the approach and accurate localization of the prey, thus completing the hunt. The bat algorithm can automatically switch from the exploration phase to the exploitation phase because exploitation becomes focused as the search approaches the global optimum in terms of changes in loudness and pulse firing rate^[47].

The movement of the virtual bat is shown below.

$$f_i = f_{\min} + (f_{\max} - f_{\min}) * \beta, \quad (13)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_{\text{best}}) * f_i, \quad (14)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (15)$$

where f define the frequency with which bats find prey, $f_{\min}$ and $f_{\max}$ are the minimum and maximum values, respectively. x_i denotes the position of the i bat in the solution space, v_i denotes the bat's speed, t describe the number of current iterations, β define a uniformly distributed random number, and $0 \leq \beta \leq 1$, x_{best} denotes the global optimal solution found so far for the entire population.

Miniature bats transmit a frequency, a fixed frequency $f_{\min}$, a variable wavelength λ , and a loudness A_0 to search for prey. The bats fly randomly at speed v_i at the position x_i . They can adjust the frequency of the emitted pulses and adjust the pulse emission rate r_i from 0 to 1 depending on the proximity of the target. in each iteration, a random number 1 is generated and compared r_i . If the random number 1 is greater than r and the search type is a local range, the target range will be further reduced, making the search more efficient and the population

update is achieved by Equation (16). In Equation (16), x_{old} denotes the best solution for the current state, δ is a random number that represents the average loudness of all bats at the current time step t , and $0 \leq \delta \leq 1$; A^t is the loudness of the t iteration.

$$x_{new} = x_{old} + \delta A^t. \quad (16)$$

Firstly, the exact position of the i individual is obtained and the fitness value F_{new} is calculated. Based on the fitness value and loudness, it is decided whether the position is updated or not, and if it needs to be updated, it can be replaced directly with the updated condition: $rand2A_i, F_{new}Fitness(x_i)$. The update of A_i and r_i is shown by Equations (17) to (18).

$$A_i^{t+1} = \alpha A_i^t, \quad (17)$$

$$r_i^{t+1} = r_i^0(1 - e^{-\gamma t}), \quad (18)$$

where α and γ are constants and are usually set to $\alpha = \gamma = 0.9$.

In the final stage of iterative processing, as the bat populations all gather toward the optimal bats, the original bat population will update to the optimal state, and the flight speed update drops to zero, resulting in a reduction in the diversity of the population and a local optimum, at which point BA is unable to continuously optimize other regions and is in a state of no evolution, reducing the overall optimization-seeking accuracy of the algorithm, and more seriously, the algorithm is difficult to solve the optimization-seeking problem in the processing of high-dimensional complex functions.

3.3 Particle Swarm Bat Fusion Algorithm

From the above analysis of the bat algorithm, we can see that Equations (14) to (15) exhibit the characteristics of fast convergence in the early stage, robustness, and easy implementation, but still suffer from the shortcomings of poor convergence accuracy in the late stage and the tendency to fall into local optimality. On this basis, the traditional bat algorithm is improved by combining the characteristics of UAV trajectory planning, and the particle swarm algorithm is integrated into the late search process of the bat algorithm. Particle swarm algorithms have the advantages of stochastic global search and local depth search, optimization, and independence from specific problems and potential parallel operations, and are therefore often used in engineering applications for function optimization, pattern classification, neural network training, fuzzy control, and other fields.

The particle swarm bat fusion algorithm is based on the bat algorithm and fuses the particle swarm optimization algorithm. Each algorithm is run through independent optimizations, i.e. particle swarm algorithms have their individuals and better solutions to replace the inferior bat individuals in the bat algorithm. Based on the position and velocity parameters, an update system that matches each other between the two is constructed, and when the algorithm performs iterative operations, it will use the particle swarm algorithm to generate better bat individuals and the traditional bat algorithm randomly generated bat individuals to compete again, select the corresponding individuals with smaller fitness values as the bat position after local search and record, then determine whether the bat position and fitness value need to be

updated, last but not least, compare the global fitness value to the current fitness value. the need for updating and replacing the current global fitness value and the ideal location of each bat within the global range, and accept synchronous update processing. Description of particle swarm bat fusion algorithm.

The specific steps for the implementation of the PSOBA are as follows.

Step 1: Parameter initialization. Set the initial value of population size n , the maximum number of iterations $NC_{\max}$, set the value of acceleration factor c_1 and c_2 , the maximum speed $v_{\max}$, the inertia value ω , the loudness A_0 , the pulse frequency r_0 , the maximum value of pulse frequency $NC_{\max}$, and the minimum value of pulse frequency $f_{\min}$.

Step 2: Randomly initialize bats. Set the i bat as $X_i = (X_1, X_2, \dots, X_n)$, according to the best position selection judgment expression $Fitness(X_i)$, get the current position of the bat, and iteratively process step by step to find out the best position in the global range (denoted by X^*) and the best $Fitness(X)$ (denoted by $F_{\min}$).

Step 3: Update the spatial location of bats. Based on the updating algorithm in BA algorithm, the corresponding data are solved by Equations (14) and (15) to do the updating process.

Step 4: Generate a random number $rand1$, if $rand1 > r_i$, then.

① The optimal individual corresponding to it is generated by Equation (16), and the fitness value, i.e. $fitba$, which is the alternative optimal position for the current local search, is also computed.

② Particle swarm operation is performed based on Equations (11) and (12) to find out the best particles and do BA local search process. In the particle swarm algorithm, its initial population, initial fitness i.e. bat population and the fitness corresponding to it, the initial population updates the initial velocity and position of the particle swarm in accordance with Equations (11) and (12) to form the optimal particle. Find the optimal particle and calculate the value of the fitness, denoted as $fitpsoba$.

③ Comparing $fitba$ and $fitpsoba$, the minimum value is noted as F_{new} , the corresponding individual with the smaller fitness value is selected as the bat location after the local search and recorded.

Step 5: Generate random number $rand2$ and compare with A_i to determine whether it needs to be updated or not, if the update condition is: $F_{\text{new}} < Fitness(X_i)$ and $rand2 < A_i$, then it needs to update the corresponding bat position and fitness value, otherwise the original position and fitness value will be kept.

Step 6: Compare the current fitness value F_{new} with the global fitness value $F_{\min}$ to determine whether the current fitness value $F_{\min}$ and X^* needs to be updated and replaced, the judgment conditions are as follows: $F_{\min} \Rightarrow F_{\text{new}}$, when both need to be updated, and accept the synchronized update process.

Step 7: Judge whether the number of searches reaches the maximum, if it is consistent with the corresponding requirements, then proceed to the next step, if not, then go back to Step 3.

The fusion algorithm still has rich population diversity in the later stage, solves the problem of precocious maturity, improves the local random search ability, allows the bat population to simulate the method of bird foraging in nature to generate new positions, and then finds the best

position and speed, thereby identifying the global best state, to make up for the shortcomings in the bat algorithm and particle swarm algorithm, and optimize the algorithm. The particle swarm algorithm is used to speed up the iterative process of finding the ideal value by applying the local search of the bat method while fully utilizing both the local and global solutions. Algorithm 1 details the pseudocode of PSOBA.

Table 1 Pseudocodes for PSOBA

Algorithm 1 PSOBA	
1.	Begin
2.	Initialize $n, f_{\max}, f_{\min}, NC_{\max}, r_0, A_0, v_{\max}, \omega, c_1, c_2$
3.	for $= 1 : n$ do
4.	for $= 1 : NC_{\max}$ do
5.	initialize the bat randomly. Let the i bat be $X_i = (X_1, X_2, \dots, X_n)$
6.	generate $Fitness(X_i)$ by Equation (8)
7.	generate $Fitness(X)$ (i.e. $F_{\min}$) and the best position X^* and assign it
8.	generate initial solution by Equations (14) and (15)
9.	if ($rand1 > r_i$) then
10.	$x_{\text{new}} = x_{\text{old}} + \delta A^t$
11.	calculate $fitba$
12.	calculate $fitpsoba$ by Equations (11) to (12)
13.	calculate F_{new} by comparing $fitba$ and $fitpsoba$
14.	end if
15.	if ($F_{\text{new}} < Fitness(X_i)$ and $rand2 < A_i$) then
16.	calculate F_{new}
17.	end if
18.	if ($F_{\min} \Rightarrow F_{\text{new}}$) then
19.	calculate $F_{\min}$ and X^*
20.	end if
21.	end for
22.	end for
23.	End

4 Results and Discussions

4.1 Parameters Setting

During the simulation trials, we used a 64-bit Windows 10 computer with an Intel(R) Core(TM) i7-8550U CPU running at 1.80 GHz or 1.99 GHz. By contrasting the performance of the various algorithms in the trajectory planning task of the UAV built in this work, the effectiveness of the enhanced particle swarm bat method is demonstrated.

The coordinates of the start point and the target point in the 3D environment simulation are (0, 0, 0) and (100, 98, 135). The parameter settings of the traditional bat algorithm, the differential evolution-based bat algorithm, and the improved particle swarm bat algorithm in this paper are shown in Table 2.

Table 2 Table of algorithm parameters

Parameter	Implication	Value
w_1	Fuel cost factor	0.2
w_2	Flight altitude cost factor	0.2
w_3	Collision threat cost factor	0.3
p	Total number of discrete points flown by drones	50
n	Particle population size	100
$H_{\min}$	A minimum safe distance from the surface	0.5
NC	Number of iterations	50
r_i^0	Initial transmitting frequency	0.3
A_i^0	Initial Loudness	0.97

4.2 Analysis of Results

Based on the above conditions, simulations of UAV trajectory planning were carried out using the traditional bat algorithm, the differential evolution-based bat algorithm, and the improved particle swarm bat algorithm in this paper, with 100 simulations. The average time taken to search the trajectory, the length of the trajectory, and the fitness value of each of the three algorithms was recorded.

The UAV trajectory planning problem in a 3D environment is different from the UAV trajectory planning problem in a 2D environment. In addition to the flight track length, the flight altitude is also an important factor in evaluating the planning performance of the algorithm. In a complex flight environment such as a mountainous city, changes in flight altitude can cause the UAV to consume more energy than if it were flying steadily over a plain. These factors are therefore considered in the cost function fitness values for further analysis.

In the graph of simulation results, irregularly shaped objects represent mountains, cubes represent residential living areas, spikes represent no-fly areas, burgundy solid dots represent PSOBA planning paths, orange hollow dots represent Differential Evolution based Bat Algorithm (DEBA) planning paths and grey solid dots represent BA planning paths. To enable the UAV to avoid threats as quickly as possible during flight, reduce flight time in complex terrain and reduce flight risk, it is necessary to shorten the flyable range^[48]. The picture makes it clear that the trajectories predicted by all three algorithms enable the UAV to reach the destination position safely from the starting point while avoiding no-fly zones and hitting any structures or mountains. When comparing the three trajectories, the PSOBA path is smoother than the DEBA and BA paths, and the trajectory is shorter.

This is a display of the simulation's results. The simulation results' front perspective is depicted in Figure 4. Figure 5 displays the simulation results from the side.

From the iterative convergence curves, we can see that after the same iterations, the convergence value of PSOBA is always lower than that of DEBA and BA, and after tens of iterations, the objective function finally converges to a relatively stable value. The lower the value of convergence adaptation, the shorter the corresponding flight path length, the lower the average flight altitude, and the lower the fuel cost.

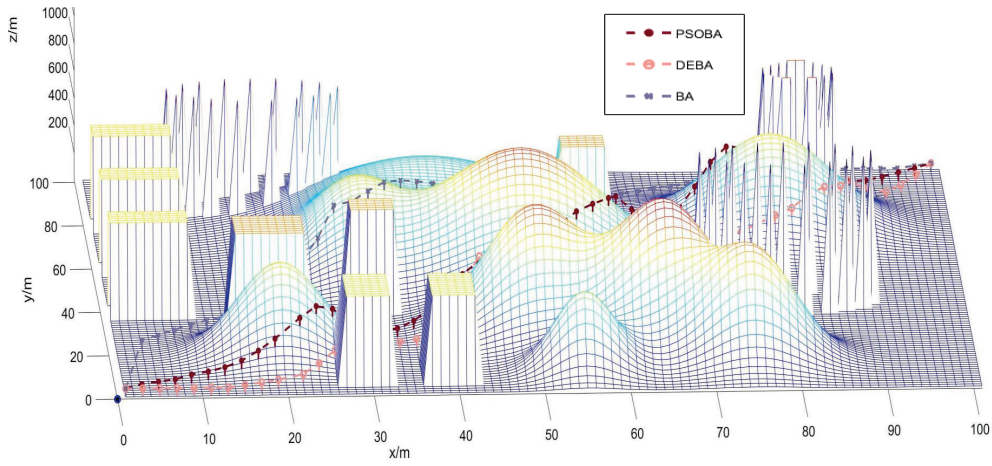


Figure 4 Front view of simulation results

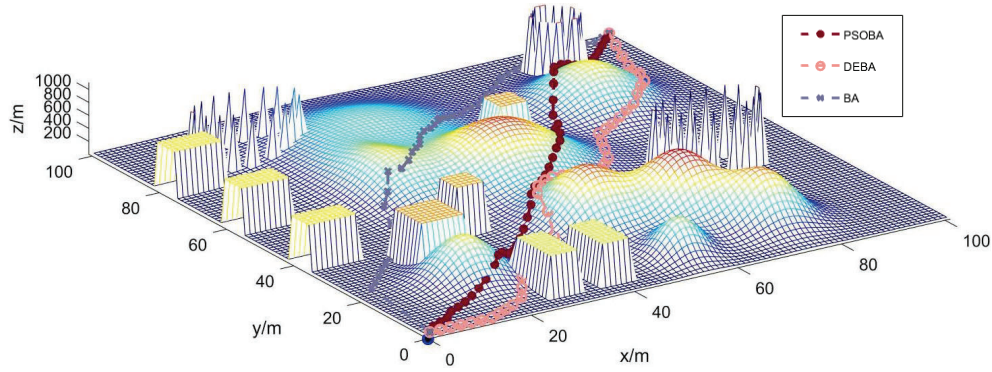


Figure 5 Overhead view of simulation results

The iterative convergence curve is shown as Figure 6.

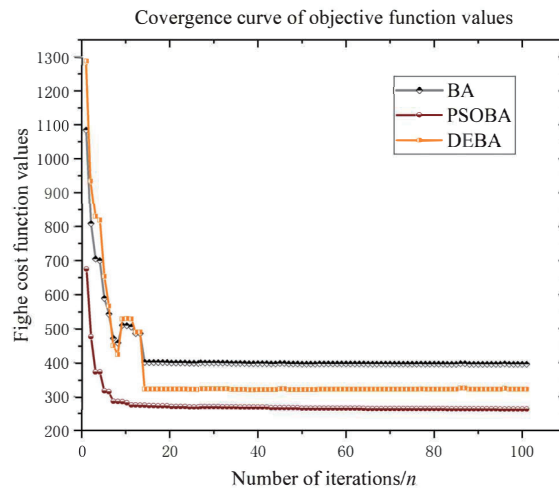


Figure 6 Convergence curve of the objective function in a 3D environment

Due to the stochastic nature of the BA algorithm search process, just one run is not sufficient to evaluate the performance of different BA algorithms. To obtain relatively reliable data and to better demonstrate the advantages of the proposed algorithms, several iterative simulations of several algorithms were therefore performed to ensure reliability. Table 3 lists the evaluation metrics for BA, DEBA, and PSOBA in a 3D environment in terms of optimal track length, flight cost fitness function value, number of final iterations, and algorithm execution time.

Figure 6 shows that the late adaptation value varies very little and does not differ significantly from the final adaptation value, despite Table 3 indicating that the enhanced PSOBA method in this article has an average of more final iterations than the DEBA algorithm. In addition, the remaining indicators of the PSOBA algorithm are significantly better than those of DEBA, thus it can be proved that the paths planned by the algorithm in this paper have better stability. The convergence values of the objective function show that the PSOBA algorithm is 23% lower than DEBA and 38% lower than BA. Comparing the algorithm execution time, the value of the PSOBA algorithm is 13.89% faster than DEBA and 20.43% faster than BA. This also shows that the flight trajectory planned in this paper is safer and has a shorter travel time.

Table 3 Table of three-dimensional environmental evaluation indicators

type of algorithm	average track length/m	the average fitness function value	the average final number of iterations possible	average algorithm execution time/s
BA	152.36	394.472	87	4.21
DEBA	160.12	317.612	84	3.89
PSOBA	175.89	244.332	87	3.35

In summary, the PSOBA algorithm can generate trajectory routes that converge faster and at a lower combined cost, which can assist police officers in reducing the time lag in understanding the conditions at the incident scene and thus performing their tasks more efficiently.

4.3 Simulation Results and Analysis Based on the Real Environment

The mountains, buildings, and no-fly zones in the 3D environment modeled using MATLAB are artificially arranged and are not always universal. Therefore this paper uses a police UAV with the primary model DJI Warp M300 RTK to take a remotely sensed terrain map of an area, then integrates this terrain map using Pix4D mapper software, followed by a DEM visualization operation of the terrain map using Global Mapper 20 software, and finally performs data extraction to plan the UAV's trajectory in the visualized terrain map. In Figure 7, the precise operating process is depicted. The PSOBA algorithm can also be verified and tested again in the simulation based on the real environment, considering that the environment changes when the police UAV is on the mission. The parameter settings for this experiment are all the same as above.

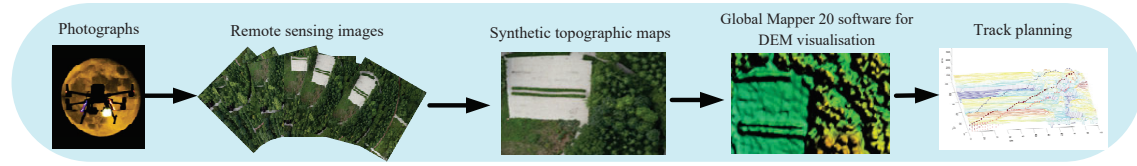


Figure 7 Operational flow based on real environment simulation

The PSOBA-planned flight route enables the UAV to safely travel from the starting point to the target location on a terrain map based on the actual surroundingst, but the flight paths planned by DEBA and BA cannot avoid the existence of breakpoints, which is not allowed to exist when the police UAV is performing its mission. By comparing the three trajectories, it is clear that the PSOBA algorithm's planned trajectory is smoother and shorter than the trajectories planned by DEBA and BA.

The simulation results based on the real environment are shown below. Figure 8 shows the side view of the simulation results. Figure 9 shows the top view of the simulation results.

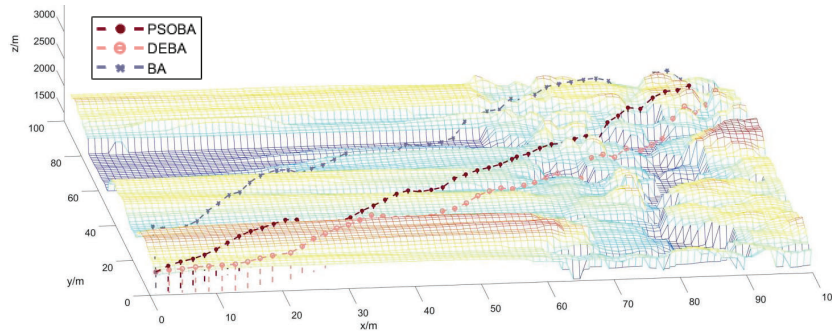


Figure 8 Side view of simulation results

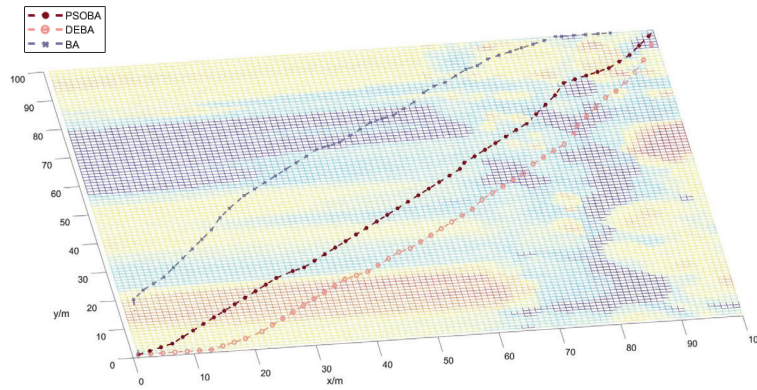


Figure 9 Top view of simulation results

From the iterative convergence curves, it can be concluded that after the same iterations, the final convergence value of PSOBA is significantly lower than that of DEBA and BA, and this convergence value is very low. The convergence curves are shown in Figure 10.

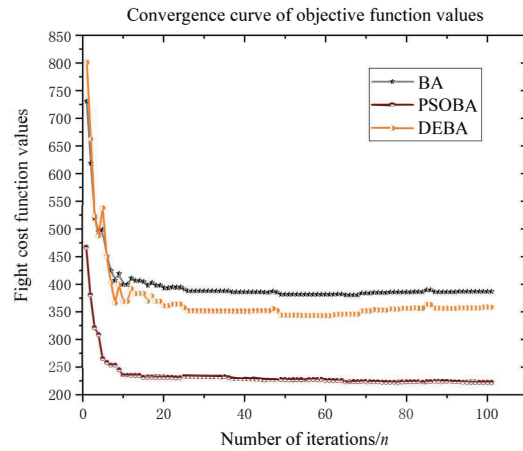


Figure 10 Convergence curve of the objective function in a 3D environment

Again, due to the random nature of the search process of the BA algorithm, just one run is not sufficient to evaluate the performance of different BA algorithms. To obtain relatively reliable data and to better demonstrate the advantages of the proposed algorithms, several iterative simulations of several algorithms were therefore carried out to ensure reliability. Table 4 lists the evaluation metrics for BA, DEBA, and PSOBA in a 3D environment in terms of optimal track length, flight cost fitness function value, number of final iterations, and algorithm execution time.

Table 4 shows that even though the improved PSOBA algorithm has an average final iteration count that is nine times higher than that of DEBA and BA, it can be seen from Figure 8 that the late adaptation value changes very little and does not differ much from the final adaptation value, and the adaptation value of the PSOBA algorithm is significantly lower than that of DEBA and BA. In addition, the rest of the indicators of the PSOBA algorithm are significantly better than those of DEBA, thus proving that the flight paths planned by this algorithm are more feasible. From the convergence values of the objective function, it can be concluded that the PSOBA algorithm has a value 40.25% lower than DEBA and 47.78% lower than BA. Comparing the algorithm execution time, the value of the PSOBA algorithm is 10.82% faster than DEBA and 24.5% faster than BA. This also indicates that the flight trajectory planned in this paper is more efficient and has a shorter travel time.

Table 4 Table of three-dimensional environmental evaluation indicators

type of algorithm	average track length/m	the average fitness function value	the average final number of iterations possible	average algorithm execution time/s
BA	163.26	394.288	81	4.04
DEBA	157.82	344.565	81	3.42
PSOBA	148.39	205.910	100	3.05

In summary, the PSOBA algorithm plans flight trajectories that are more suitable for the changing environment in which police UAVs fly and can produce perfect flight trajectories with a lower overall cost.

5 Summary and Outlook

In an asymmetric network architecture environment like mountain cities, intelligent control of UAVs has been a hot topic for the trajectory planning problem of police UAVs. This study employs a 3D grid topographic map to accurately represent the landscape of a mountainous city in order to imitate the real situation. It also establishes a cost function to attempt to recreate the scenario of a police UAV on a mission. The particle swarm optimization algorithm is grafted onto the bat algorithm to effectively solve the problems of low convergence accuracy, the classic bat technique has a slow convergence rate and a propensity to settle into local minima, while B-sample curves can be used to improve the stability of the UAV flight path. This reduces the amount of time needed to plan the flight path. The PSOBA algorithm outperforms DEBA and BA and greatly raises the success rate of UAV trajectory planning, according to experimental findings and objective analysis. The convergence time is reduced, which is undoubtedly more appropriate for police UAVs applied in special scenarios.

Although this paper makes targeted improvements to the trajectory planning time of police UAVs to improve the speed of trajectory planning and obtain lower-cost trajectories, there is still much room for improvement in the field of mobile intelligent device trajectory planning research. UAVs occasionally need to fly near the ground in the process of searching for targets, when there are dynamic threats or local bad weather, and thus online dynamic UAV track planning is the next research direction in order to cope with the dynamically changing 3D complex environment.

References

- [1] Shahid N, Abrar M, Ajmal U, et al. Path planning in unmanned aerial vehicles: An optimistic overview. *International Journal of Communication Systems*, 2022, 35(6): e5090.
- [2] Qadir Z, Ullah F, Munawar H S, et al. Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review. *Computer Communications*, 2021, 168: 114–135.
- [3] Aggarwal S, Kumar N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, 2020, 149: 270–299.
- [4] Huang Z, Chen C, Pan M. Multiobjective UAV path planning for emergency information collection and transmission. *IEEE Internet of Things Journal*, 2020, 7(8): 6993–7009.
- [5] Puente-Castro A, Rivero D, Pazos A, et al. A review of artificial intelligence applied to path planning in UAV swarms. *Neural Computing and Applications*, 2022, 324(1): 153–170.
- [6] Qu C, Gai W, Zhang J, et al. A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. *Knowledge-Based Systems*, 2020, 194: 105530.
- [7] Wan Y, Zhong Y, Ma A, et al. An accurate UAV 3-D path planning method for disaster emergency response based on an improved multiobjective swarm intelligence algorithm. *IEEE Transactions on Cybernetics*, 2022, 53(4): 2658–2671.
- [8] Khan M T R, Muhammad Saad M, Ru Y, et al. Aspects of unmanned aerial vehicles path planning: Overview and applications. *International Journal of Communication Systems*, 2021, 34(10): e4827.
- [9] Coombes M, Fletcher T, Chen W H, et al. Optimal polygon decomposition for UAV survey coverage path planning in wind. *Sensors*, 2018, 18(7): 2132.

- [10] Li D, Yin W, Wong W E, et al. Quality-oriented hybrid path planning based on a* and q-learning for unmanned aerial vehicle. *IEEE Access*, 2021, 10: 7664–7674.
- [11] Ambrosino G, Ariola M, Ciniglio U, et al. Path generation and tracking in 3-D for UAVs. *IEEE Transactions on Control Systems Technology*, 2009, 17(4): 980–988.
- [12] Wang G, Guo L, Duan H, et al. Path planning for uninhabited combat aerial vehicle using hybrid meta-heuristic DE/BBO algorithm. *Advanced Science, Engineering and Medicine*, 2012, 4(6): 550–564.
- [13] Klauser F. Police drones and the air: Towards a volumetric geopolitics of security. *Swiss Political Science Review*, 2021, 27(1): 158–169.
- [14] Enemark C. Armed drones and ethical policing: Risk, perception, and the tele-present officer. *Criminal Justice Ethics*, 2021, 40(2): 124–144.
- [15] Klauser F. Policing with the drone: Towards an aerial geopolitics of security[J]. *Security Dialogue*, 2022, 53(2): 148–163.
- [16] Aiello G, Valavanis K P, Rizzo A. Fixed-Wing UAV energy efficient 3D path planning in cluttered environments. *Journal of Intelligent & Robotic Systems*, 2022, 105(3): 60.
- [17] Zheng X, Wang F, Xia J, et al. The methodology of UAV route planning for efficient 3D reconstruction of building model. 2017 25th International Conference on Geoinformatics, Buffalo, NY, USA. IEEE, 2017: 1–4. doi: 10.1109/GEOINFORMATICS.2017.8090905.
- [18] Zhou X, Gao F, Fang X, et al. Improved bat algorithm for UAV path planning in three-dimensional space. *IEEE Access*, 2021, 9: 20100–20116.
- [19] Chen X, Chen X, Xu G. The path planning algorithm studying about UAV attacks multiple moving targets based on Voronoi diagram. *International Journal of Control and Automation*, 2016, 9(1): 281–292.
- [20] Rivera N, Hernandez C, Hormazbal N, et al. The 2^k neighborhoods for grid path planning. *Journal of Artificial Intelligence Research*, 2020, 67: 81–113.
- [21] Wu X, Xu L, Zhen R, et al. Biased sampling potentially guided intelligent bidirectional RRT? algorithm for UAV path planning in 3D environment. *Mathematical Problems in Engineering*, 2019, 2019: 1–12.
- [22] Huang J, Sun W. A method of feasible trajectory planning for UAV formation based on bi-directional fast search tree. *Optik*, 2020, 221: 165213.
- [23] Bai X, Jiang H, Cui J, et al. UAV path planning based on improved A* and DWA algorithms. *International Journal of Aerospace Engineering*, 2021. doi: 10.1155/2021/4511252.
- [24] Xiao S, Tan X, Wang J. A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction. *Electronics*, 2021, 10(7): 853.
- [25] Huang C, Fei J. UAV path planning based on particle swarm optimization with global best path competition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2018, 32(6): 1859008.
- [26] Zhang Z, Lu R, Zhao M, et al. Robot path planning based on genetic algorithm with hybrid initialization method. *Journal of Intelligent & Fuzzy Systems*, 2022, 42(3): 2041–2056.
- [27] Wang G G, Chu H C E, Mirjalili S. Three-dimensional path planning for UCAV using an improved bat algorithm. *Aerospace Science and Technology*, 2016, 49: 231–238.
- [28] Lin N, Tang J, Li X, et al. A novel improved bat algorithm in UAV path planning. *Computers, Materials & Continua*, 2019, 61(1): 323–344.
- [29] Zhang H, Zhuang Q, Li G. Robot path planning method based on indoor spacetime grid model. *Remote Sensing*, 2022, 14(10): 2357.
- [30] Maboudi M, Homaei M R, Song S, et al. A review on viewpoints and path planning for UAV-based 3D reconstruction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023, 16: 1–26.
- [31] Han B, Qu T, Tong X, et al. Grid-optimized UAV indoor path planning algorithms in a complex environment. *International Journal of Applied Earth Observation and Geoinformation*, 2022, 111: 102857.
- [32] Wang G, Guo L, Duan H, et al. A bat algorithm with mutation for UCAV path planning. *The Scientific World Journal*, 2012. doi: 10.1100/2012/418946.
- [33] Chowdhury A, De D. RGSO-UAV: Reverse glowworm swarm optimization inspired UAV path-planning in a 3D dynamic environment. *Ad Hoc Networks*, 2023, 140: 103068.
- [34] Wang H, Tan L, Shi J, et al. An improved NSGA-II algorithm for UAV path planning problems. *Journal of Internet Technology*, 2021, 22(3): 583–592.
- [35] Ferrandez S M, Harbison T, Weber T, et al. Optimization of a truck-drone in tandem delivery network

- using k-means and genetic algorithm. *Journal of Industrial Engineering and Management (JIEM)*, 2016, 9(2): 374–388.
- [36] Wu Y, Low K H, Lv C. Cooperative path planning for heterogeneous unmanned vehicles in a search-and-track mission aiming at an underwater target. *IEEE Transactions on Vehicular Technology*, 2020, 69(6): 6782–6787.
- [37] Huang S, Tian J, Qiao L, et al. Unmanned aerial vehicle path planning based on improved genetic algorithm. *Journal of Computer Applications*, 2021, 41(2): 390.
- [38] Eshtehardian S A, Khodaygan S. A continuous RRT*-based path planning method for non-holonomic mobile robots using B-spline curves. *Journal of Ambient Intelligence and Humanized Computing*, 2023, 14(7): 8693–8702.
- [39] Bi Q Z, Huang J, Lu Y A, et al. A general, fast and robust B-spline fitting scheme for micro-line tool path under chord error constraint. *Science China Technological Sciences*, 2019, 62: 321–332.
- [40] Freitas D, Lopes L G, Morgado-Dias F. Particle swarm optimisation: A historical review up to the current developments. *Entropy*, 2020, 22(3): 362.
- [41] Zhang X, Lin Q. Three-learning strategy particle swarm algorithm for global optimization problems. *Information Sciences*, 2022, 593: 289–313.
- [42] Gad A G. Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering*, 2022, 29(5): 2531–2561.
- [43] Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia. IEEE, 1995: 1942–1948. doi: 10.1109/ICNN.1995.488968.
- [44] Agarwal T, Kumar V. A systematic review on bat algorithm: Theoretical foundation, variants, and applications. *Archives of Computational Methods in Engineering*, 2022, 29(5): 2707–2736.
- [45] Al-Dyani W Z, Ahmad F K, Kamaruddin S S. Improvements of bat algorithm for optimal feature selection: A systematic literature review. *Intelligent Data Analysis*, 2022, 26(1): 5–31.
- [46] Yang X S. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, Berlin, Heidelberg, 2010: 65–74.
- [47] Alyasseri Z A A, Alomari O A, Al-Betar M A, et al. Recent advances of bat-inspired algorithm, its versions and applications. *Neural Computing and Applications*, 2022, 34(19): 16387–16422.
- [48] Huang R, Tian J. Wavelet-based elman neural network with the modified differential evolution algorithm for forecasting foreign exchange rates. *Journal of Systems Science and Information*, 2021, 9(4): 421–439.