

## Research on Hybrid Path Planning Algorithms for UAVs in Complex Environments

Xinyue Chang<sup>1</sup>, Liang Ye<sup>1,\*</sup>, Lin Ma<sup>1</sup> and Shuyi Chen<sup>1</sup>

<sup>1</sup> Harbin Institute of Technology, Harbin, China

### Abstract

**INTRODUCTION:** This paper investigates a UAV path planning algorithm in a UAV-assisted network scenario, integrating both global and local path planning. Firstly, the ASPSO (Adaptive Spherical Vector-Based Particle Swarm Optimization) algorithm is proposed for offline path planning to obtain key global path points, providing a general flight strategy for the UAV. During the flight, the UAV continuously detects surrounding obstacles in real-time. If newly detected obstacles are encountered, the ECAVF (Enhanced Collision Avoidance Vector Field) algorithm is employed for local path planning to dynamically avoid obstacles and ensure the safety of the UAV.

**OBJECTIVES:** The objective of this paper is to enhance the path planning capability of existing algorithms in complex three-dimensional environments, enabling UAVs to operate efficiently and safely.

**METHODS:** The proposed ASPSO algorithm determines parameter ranges for different scenarios during the initialization phase, effectively reducing initialization time. Additionally, a multi-strategy optimization approach is introduced during the search process. Expanding the search space in the early iterations helps escape local optima, while minor perturbations are introduced in the later iterations to continue exploring within the neighbourhood of high-quality solutions. Finally, a method utilizing virtual control points for path refinement is proposed to smooth the trajectory. The ECAVF algorithm incorporates a dynamic adjustment factor based on relative velocity to optimize the vector field in the presence of multiple moving obstacles. By integrating factors such as distance and velocity, a hybrid vector field is constructed, demonstrating superior robustness in complex multi-obstacle scenarios.

**RESULTS:** The proposed method is compared with the PSO (Particle Swarm Optimization), the Spherical Vector-based PSO, and the original CAVF (Collision Avoidance Vector Field) method. The results demonstrate that the proposed method exhibits higher initialization efficiency, superior initial solution quality, and the ability to obtain a more optimal global path. Additionally, it shows stronger dynamic obstacle avoidance capabilities and a higher success rate in avoiding obstacles.

**CONCLUSION:** These results demonstrate that the proposed method effectively enhances the quality of global path planning solutions and improves the success rate of dynamic obstacle avoidance.

**Keywords:** UAV Path Planning, ASPSO, ECAVF

Received on 27 March 2025, accepted on 19 May 2025, published on 27 May 2025

Copyright © 2025 X. Chang *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetsis.8974

### 1. Introduction

Unmanned Aerial Vehicles (UAVs) are a type of flying mobile robots capable of performing various tasks and operating in dynamic environments. Advances in technology and reduced unit costs have expanded the application of UAVs in civilian markets. Due to their

flexibility, manoeuvrability, and the ability to equip them with various sensors, transmission devices, and visual capture equipment, UAVs have found widespread use in multiple fields. UAVs have strong manoeuvrability [1], and with the advancement of technology, they are used in detection, and search and rescue. UAV path planning needs to consider task time, flight distance, and safety, and UAVs may encounter many obstacles during flight, such as

\*Corresponding author. Email: yeliang@hit.edu.cn

buildings, mountains, radar areas, etc. The path of UAVs is research hotspot, and it is necessary to design the path under various constraint conditions, fulfil task [2]. Path planning for UAVs is one of the critical challenges that need to be addressed, encompassing both global path planning and local path planning. Global path planning, also known as static planning, requires prior knowledge of the environment. It involves planning a path between a given start point and target point in a fixed environment, aiming to find an optimal or feasible path that allows the UAV to reach the destination with minimal cost. Local path planning, on the other hand, focuses on real-time, online planning in partially unknown or dynamically changing environments to adapt to various dynamic changes in uncertain conditions. With the continuous research in this field, scholars have proposed many methods. Traditional search methods include A\* algorithm, Fast RRT [3], and Artificial Potential Field (APF). However, these methods have slow convergence speed. Therefore, swarm intelligence algorithms have been proposed, including genetic algorithm (GA) [4], grey wolf algorithm (GWO) [5], ant colony optimization(ACO) [6], particle swarm optimization (PSO) [7] and other ways. A survey [8] proposes a new framework for UAV planning using an improved PSO algorithm. Improved PSO algorithm applied to get the best path. A survey [9] proposes a path design and energy control based on the prediction of user movement information. Advocating a Three-Phase Machine Learning Process. Get user's location and path design of UAVs. Many UAVs serve as agents that learn the best action by communicating with the environment. In PSO, because the update of particle relies on the global optimal one. A survey [10] proposes SDPSO way, it binding the update of the PSO global solution with SA. It can to improve the diversity of the method. The particle diversity's lack can make it fall into local optima easily. Due to poor detection ability and lack of particle diversity of PSO, they are easily affected by local. In survey [11], a reconfigurable intelligent surface (RIS) assisted UAV relay communication system is proposed, optimizes the drone's flight path, RIS's passive beamforming, and power distribution to maximize downlink throughput.

## 2. System Model

### 2.1. System Model

Our geomorphological data in real digital elevation model is obtained and the higher natural mountains in the flight environment are constructed on its basis. The mountain information is described by the function:

$$Peak(x, y) = \sum_{i=1}^n h_i \times \exp\left(-\left(\frac{x-x_{ic}}{x_s}\right)^2 - \left(\frac{y-y_{ic}}{y_s}\right)^2\right) \quad (1)$$

The  $n$  is peaks amount, the peak center  $i$  is  $(x_{ic}, y_{ic}, 0)$ ,  $x_s$  and  $y_s$  is the attenuation of the height, denoted as the slope. Threats such as buildings, radar detection, extreme

weather. May also be encountered during the flight, here, these threats are abstractly represented as cylinders. The coordinates of the circle centre in the horizontal plane of cylinder  $i$  is  $(x_i, y_i, 0)$ . Figure 1 is the environment.

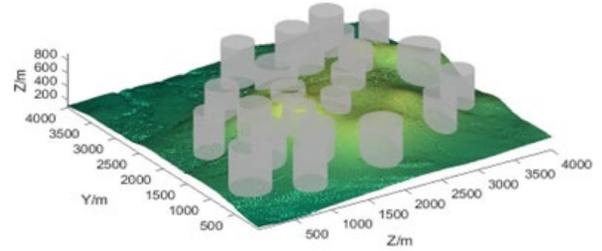


Figure 1. Flight Environment

The communication link from UAV to user can be considered as an air to ground transmission [12]. It is assumed that the UAV to user link experiences LoS and NLoS are random. The probability of LoS occurrence for this link is provided as follows:

$$P_{LoS}(\theta_t) = b_1 \left( \frac{180}{\pi} \theta_t - \zeta \right)^{b_2} \dots \dots \dots (2)$$

The  $\theta_t = \sin^{-1}(h(t)/d(t))$ . UAV's altitude above ground is  $h(t)$  and the distance of link is  $d(t)$ . The  $b_1$  and  $b_2$  are fixed, which are impacted by environment. NLoS probability is given by  $P_{NLoS} = 1 - P_{LoS}$ . The power gain can be calculated as

$$g(t) = K_0^{-1} d^{-2} [P_{LoS} \mu_{LoS} + P_{NLoS} \mu_{NLoS}]^{-1} \quad (3)$$

Where  $K_0 = (4\pi f_c / c)^2$ , defined  $\alpha$  as path loss paramant. The  $f_c$  is carrier frequency and  $c$  is light speed.  $r(t)$  is the user's rate at time  $t$

$$r(t) = B \log_2 \left( 1 + \frac{p(t)g(t)}{\sigma^2} \right) \dots \dots \dots (4)$$

Where  $\sigma^2 = BN_0$  with  $N_0$  is the power spectral density of the additive WGN.  $p(t)$  is UAV transmission power,  $g(t)$  is UAV to user channel gain

### 2.2 Problem Formulation

In order to maintain efficient and safe UAV flight in complex environments, the path should be optimal by multiple constraints in the actual environment. The UAV path is controlled by the key points on the path, a UAV flight path  $T$ ,  $C$  represents the set of  $n$  necessary path point positions, and  $C_i = (x_i, y_i, z_i)$ . The distance between the two control points is calculated using the Euclidean metric, and path length cost is calculated as

$$F_1(T) = \sum_{i=1}^{n-1} \|\vec{C}_i C_{i+1}\| \quad (5)$$

UAVs are also constrained in terms of flight height, with minimum and maximum height limits between the node and the ground surface being  $a_{\min}$  and  $a_{\max}$ , respectively.

$\Delta a = a_{\max} - a_{\min}$ , the optimum height of the flight in relation to the surface of the obstacle is  $a_{\text{best}} = \frac{a_{\min} + a_{\max}}{2}$ , the altitude cost is computed as

$$A_i = \begin{cases} \infty & \text{otherwise} \\ -\frac{\Delta a}{2} \times \log\left(\frac{\Delta a - 2|a_i - a_{\text{best}}|}{\Delta a}\right) & a_{\min} \leq a_i \leq a_{\max} \end{cases} \quad (6)$$

Summing  $A_i$  for all points gives the height cost:

$$F_2(T) = \sum_{i=1}^n A_i \quad (7)$$

Assuming that all obstacle sets are  $O$ . The centre of its projection on the  $xoy$  plane is  $c_0$  and the radius  $R_0$ , the length of the UAV's body is  $l$ , and the dangerous distance for collision between the UAV and obstacles is  $Z$  as shown in Figure 2. The half of the central angle corresponding to the secant of circle  $C$  is denoted as  $\theta_u$ , and  $\theta_u$  is calculated as

$$\theta_u = \arccos\left(\frac{d_o}{R_0 + l + Z}\right) \quad (8)$$

To avoid collisions, the maximum value of  $\theta_u$  is  $\theta_m$ , and  $\theta_m$  is calculated as

$$\theta_m = \arccos\left(\frac{R_0 + l}{R_0 + l + Z}\right) \quad (9)$$

The ratio of the two is recorded as  $\theta_r$ , and  $\theta_r = \frac{\theta_u}{\theta_m}$ . The threat cost calculation for obstacles is as follows:

$$B(\vec{C}_i C_{i+1}) = \begin{cases} \infty & d_o < R_0 + l \\ e^{-\frac{\theta_r}{\theta_u - 1}} & R_0 + l < d_o < R_0 + l + Z \\ 0 & d_o > R_0 + l + Z \end{cases} \quad (10)$$

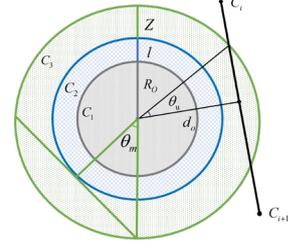


Figure 2. Obstacle Threat

The smoothness of the path is also an important metric to look at. As shown in Figure 3.

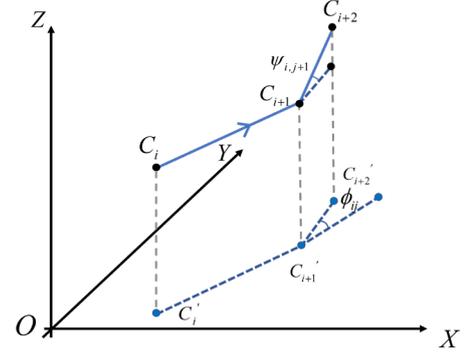


Figure 3. Path smoothness

The direction angle  $\phi_i$  and the climbing angle  $\psi_i$ , they are calculated as

$$\phi_{i,i+2} = \arctan\left(\frac{\|\vec{C}_i C_{i+1} \times \vec{C}_{i+1} C_{i+2}\|}{\|\vec{C}_i C_{i+1} \cdot \vec{C}_{i+1} C_{i+2}\|}\right) \quad (11)$$

$$\psi_i = \arctan\left(\frac{z_{i+1} - z_i}{\|\vec{C}_i C_{i+1}\|}\right) \quad (12)$$

The path smooth cost is then computed as:

$$F_4(T) = a_1 \sum_{i=1}^{n-2} \phi_i + a_2 \sum_{i=1}^{n-1} |\psi_i - \psi_{i-1}| \quad (13)$$

The overall cost is a linear weighting of the four terms, the overall cost is calculated as

$$F_{\text{sum}} = w_1 F_1(T) + w_2 F_2(T) + w_3 F_3(T) + w_4 F_4(T) \quad (14)$$

For a detected obstacle with its centroid initially at position  $P_o \in \Omega$ , its velocity is denoted as  $V_o \in \mathbb{I}^N$  if it moves. The obstacle's boundary  $\chi^b$  can be represented by variables longitude angle  $u$  and latitude angle  $v$ , where  $u \in [0, 2\pi]$  and  $v \in [0, \pi]$ , as shown in Equation (15).

$$P^b(t, u, v) = \begin{pmatrix} a(t) \cos(u) \sin(v) \\ b(t) \sin(u) \sin(v) \\ c(t) \cos(v) \end{pmatrix} \quad (15)$$

Here,  $a(t)$ ,  $b(t)$ , and  $c(t)$  are time-dependent functions representing the radii of the ellipsoidal obstacle along the x, y, and z axes, respectively. The influence distance of the obstacle is denoted as  $\rho$ .

The motion of the UAV is determined by double-integrator dynamics, where the system state encompasses the dynamic behavior of two integral variables: position and velocity.

$$\dot{\mathbf{x}} = \mathbf{v}, \dot{\mathbf{v}} = \mathbf{u} \quad (16)$$

Under the influence of double-integrator dynamics, the state of the UAV is represented by Equation (17).

$$\mathbf{x}(t) = \begin{pmatrix} P(t) \\ \mathbf{v}(t) \end{pmatrix}, \mathbf{u}(t) = \begin{pmatrix} \mathbf{v}(t) \\ \mathbf{u}(t) \end{pmatrix} \quad (17)$$

### 3 Adaptive Spherical Vector Based PSO

In this paper, ASPSO is proposed based on SPSO algorithm [13], which is improved in three aspects: initialization process, search process and path optimization.

#### 3.1 Spherical Vector based PSO

In SPSO algorithm, the location information of particle is represented as a vector containing the magnitude, azimuth angle and elevation angle. Path  $P$  includes  $n$  nodes, and  $P$  is as follows:

$$P = (r_1, \phi_1, \psi_1, r_2, \phi_2, \psi_2, \dots, r_n, \phi_n, \psi_n) \quad (18)$$

The velocity associated to that particle is defined as a vector:

$$\Delta P = (r'_1, \phi'_1, \psi'_1, r'_2, \phi'_2, \psi'_2, \dots, r'_n, \phi'_n, \psi'_n) \quad (19)$$

The position and velocity information of each node can be represented  $u_{ij} = (\rho_{ij}, \phi_{ij}, \psi_{ij})$  and  $v_{ij} = (\Delta\rho_{ij}, \Delta\phi_{ij}, \Delta\psi_{ij})$ . The update equations for SPSO are given by:

$$v_{ij}^{t+1} = wv_{ij}^t + \eta_1\gamma_1 (P_{best}^t - u_{ij}^t) + \eta_2\gamma_2 (g_{best}^t - u_{ij}^t) \quad (20)$$

$$u_{ij}^{t+1} = u_{ij}^t + v_{ij}^{t+1} \quad (21)$$

#### 3.2 An Initialization Method Based on Environmental Information

Unlike traditional particle swarm optimization algorithms in Cartesian coordinates, each particle is independently and randomly generated without any connection to each other.

In a spherical coordinate system, there is a strong correlation between particles, and the latter particle is generated in a coordinate system with the previous particle as the origin. Therefore, there is a strong chain reaction between particles, and the flight environment in this paper is complex, which has high constraints on the UAVs.

The position information of particles consists of magnitude, azimuth angle and elevation angle. It can also be interpreted as the step size  $r$ , turning angle  $\phi_i$  and the climbing angle  $\psi_i$  during the search. Due to the uneven terrain, there may be a significant height difference between the starting and ending points. For the climbing angle, a larger range may directly miss the optimal solution, while a smaller range may not be able to adapt to the undulating changes of the terrain. Therefore, it is necessary to set the range of climbing angle changes for different terrain information.

Start point is  $(X_s, Y_s, Z_s)$ , the terminal point is  $(X_E, Y_E, Z_E)$ ,  $Z$  indicates absolute height, the distance between the two is  $d$  and the height difference is  $h$ . The number of particles in the ASPSO algorithm is  $N$ . The maximum search step size is set to  $r_{max} = \frac{2 \cdot d}{N}$ . This is to

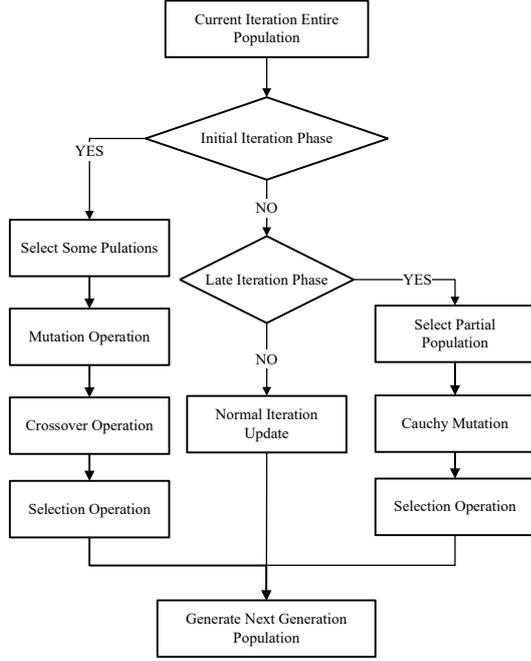
avoid skipping the optimal solution directly if the  $r$  is too large, and to avoid reducing the search ability if the  $r$  is too small. The turning angle  $\phi \in (-\pi, \pi)$ , allows particles to have the maximum search range on the horizontal plane. The setting of the climbing angle range, as follows:

$$\psi = \text{atan} \left( \frac{2 \cdot h}{r_{max}} \right) \quad (22)$$

Based on the point location, combined with the search step size, controlling climbing angle within a reasonable range can improve the initialization speed while seeking an optimal initial solution.

#### 3.3 A Multi Strategy Optimization Method

The position parameters of control points during the flight process play a decisive role in the entire path. For this purpose, a joint differential evolution multi-strategy optimization method is proposed. Considering the dense density of obstacles, in order to avoid interference from obstacles, it is need to continuously adjust the direction of flight. In the early stages of iteration, combined with the idea of differential evolution, the search range is expanded. In the late stages of iteration, for solutions with low fitness and high quality, the idea of Cauchy mutation is proposed, a small disturbance is introduced to further search in the neighborhood space of the optimal solution. The process of multi strategy optimization method as shown in Figure 4.



**Figure 4.** Multi strategy Flowchart

Due to the large population size, in order to improve running speed and reduce computational complexity, a portion of the population is selected for each iteration process. For each iteration, we refer to the selection rate function to determine the size, as shown below:

$$r_s = \frac{(r_e - r_f) \times i}{Inter} + r_f \quad (23)$$

Where  $r_f$  is the initial selection rate,  $r_e$  is the final selection rate,  $Inter$  is times,  $i$  is the current times.  $s\_num$  is the number of population,  $s\_num$  is defined as

$$s\_num = npop \cdot r_s \quad (24)$$

At the beginning of the iteration, refer to the selection rate formula and select populations. For each population, the number of control points is  $N$ , and the turning angle parameters of the control points in population  $p$  is represented as

$$phi_p = (phi_{p,1}, phi_{p,2}, \dots, phi_{p,N-1}, phi_{p,N}) \quad (25)$$

Mutation generates mutation vectors. Mutation vector is represented by  $v_{p,n}$ . The mutation strategy refers to  $DE/current$  to  $rand/1$ , but due to the limitation of the optimal solution of the early iterations, we only add random variables to the current solution as follows

$$v_p = phi_p + F \cdot (phi_{r_1} - phi_{r_2}) \quad (26)$$

Where  $r_1$  and  $r_2$  are mutually exclusive random numbers and  $F$  is scaling factor.

Crossover operation produces a random set of vectors, we use a binomial cross strategy, the formula as shown below

$$u_{p,n} = \begin{cases} v_{p,n} & rand < CR \text{ or } n = j \\ phi_{p,n} & otherwise \end{cases} \quad n = 1, 2, \dots, N \quad (27)$$

$N$  is dimensions and  $CR$  is crossover rate. The  $rand$  is a uniform distribution between 0 and 1.  $j$  is a random integer between 1 to  $N$ .

Select operation to compare the fitness of the original solution with the fitness of the solution after mutation and crossover and preserve populations with lower fitness.

In the late stages of iteration, it is determined whether the selected population fitness is in a relatively optimal state. If it is a good solution, Cauchy mutation is performed. Taking the turning angle as an example, the Cauchy mutation operation is shown as follow

$$phi_{p,n} = phi_{p,n} + k \cdot C(R_p; x_0, \gamma) \cdot phi_{p,n} \quad (28)$$

Where  $k$  is a random number from 1 to -1. Ranking all population fitness in descending order,  $R_p$  is the proportion of the fitness of the population  $p$  relative to all fitness values.  $C(R_p; x_0, \gamma)$  represents the probability distribution function of the Cauchy function. After the Cauchy mutation, fitness is calculated again, comparing the original population fitness with the population fitness, retaining the population with the lower fitness, and continuing to produce the next generation of populations.

### 3.4 A Path Optimization Strategy Based on Virtual Control Points

In a simple and flat terrain environment, the influence of terrain height can be ignored, and the constraints on the altitude of flight nodes are relatively weak. However, in this paper, due to the complex environment, dense obstacles, and high terrain fluctuations, the distance between control points in a population may be far apart. Although each individual control point satisfies the constraint conditions of flight altitude, when all control points form a continuous path, there may be collisions between the path and the ground, which directly poses a safety threat to the flight nodes. Therefore, we propose a path optimization strategy based on virtual control points. On the basis of the initial path, a single dimensional small step search method is used to find path segments that do not meet height constraints, and then combined with the terrain environment, virtual control points are constructed for Bezier curve fitting. This process optimizes the existing path while ensuring low fitness, reducing path anomalies such as emergency turns and sudden height changes.

In previous studies, scholars directly create spline interpolation based on the control points, and then obtain optimized curves based on the interpolation results or construct a Bezier curve through control points. But the

environments provide a small margin of variation space for UAVs. Blind optimization may result in trajectories colliding with obstacles.

### Single Dimensional Small Step Search Method

In practical environments, UAVs move in three-dimensional space, and the initial path is a three-dimensional path. Here, a single dimensional small step search method is proposed. For the initial path, the initial path is divided into small path segments based on control points as nodes. Gradually search along a dimension with smaller step sizes to obtain regions that do not meet the constraint conditions.

An initial path is defined as  $L_{xyz}$ , the starting point of the path is  $(X_S, Y_S, Z_S)$  and the endpoint is  $(X_E, Y_E, Z_E)$ . The points set is  $\{X_S, Y_S, Z_S; X_1, Y_1, Z_1; \dots; X_n, Y_n, Z_n; X_E, Y_E, Z_E\}$ . The path  $L_{xyz}$  is actually composed of  $L_{xyz,i}$ ,  $i = 1, 2, \dots, n, n+1$ .

Single dimension small step search method is as follows, taking the search process along the x axis direction as an example, assuming that the search step size is  $S_l$ , the distance interval in the x axis direction of each key point is expressed as follows

$$\begin{cases} \Delta x_1 = X_1 - X_S \\ \Delta x_2 = X_2 - X_1 \\ \dots \\ \Delta x_n = X_n - X_{n-1} \\ \Delta x_{n+1} = X_E - X_n \end{cases} \quad (29)$$

On this basis, the number of search steps for  $L_{xyz,i}$  is  $step_i$

and  $step_i = \left\lfloor \frac{\Delta x_i}{S_l} \right\rfloor$ .  $L_{xoy,i}$  is the horizontal projection of

$L_{xyz,i}$ . Starting from the  $L_{xoy,i}$  starting point, taking the x axis direction of the path as the exploration direction and  $S_l$  as the exploration step size. The set of x axis coordinates of all search points is  $x_{is} = \{x_{is,1}, x_{is,2}, \dots, x_{is,step_i-1}, x_{is,step_i}\}$ .

Based on the function of  $L_{xoy,i}$ , obtain the corresponding ordinate  $y_{is}$ . Based on the function of  $L_{xyz,i}$ , obtain the corresponding altitude coordinate  $z_{is}$ . Analyse whether the altitude of the search point satisfies the flight constraints of the UAV, and for the search point that does not satisfy the height constraints, construct the virtual control point with reference to the height of the terrain. The coordinates are denoted as  $(x_{iv}, y_{iv}, z_{iv})$ , and the height is set as follow

$$z_{iv} = Peak(x_{iv}, y_{iv}) + \alpha \cdot a_{best} \quad (30)$$

Where  $Peak(x_{iv}, y_{iv})$  is the terrain altitude,  $a_{best}$  is the flight optimum altitude, and  $\alpha$  is the altitude adjustment factor.

### Bezier Fitting Based on Virtual Control Points

For each small path  $L_{xyz,i}$ , a set of starting, ending, and virtual points will be obtained (if there is a case where the search points do not satisfy the height condition constraints).

The Bezier curve can be represented as a product of a basis function and a vector, as follows

$$B(t) = \sum_{i=1}^n B_{in}(t)P_i \quad (31)$$

The Bernstein basis function as follow

$$B_{in}(t) = C_n^i t^i (1-t)^{n-i}, i = 1, 2, \dots, n \quad (32)$$

In practical engineering problems, high order Bezier curves are often not used because their curvature is too high, leading to a decrease in smoothness. Therefore, the commonly used fourth order Bezier curve in engineering is used for path smoothing. The curve consists of five points. In this paper, if there are too many path nodes, the higher height nodes are prioritized as control points. If there are less than five path nodes, use a low order Bessel curve for optimization.

## 3.5 Experimental Studies and Comparative Analysis

To exam ASPSO method, it will be compared with multiple algorithms. We visualize the path during the simulation process. The three-dimensional space for flight is  $4 \text{ km} \times 4 \text{ km} \times 1 \text{ km}$ . For the transmission model, the following [12] gives the parameter settings .

Table 1. Parameters Setting

Parameter	Description	Value
$f_c$	Carrier frequency	2GHz
$P$	UAV output power	30dBm
$N_0$	Noise power spectral	-170dBm/Hz
$B$	Bandwidth	1MHz
$\mu_{LoS}$	LoS additional path loss	3dB
$\mu_{NLoS}$	NLoS additional path loss	23dB
$b_1$	Environmental parameters	0.36
$b_2$	Environmental parameters	0.21

The is 0.891 and the ending point is set directly above the user, with the UAV positioned at a height of 200 meters from the user.

### Comparative Analysis of Initialization Process

Compare and analyze the initialization iterations times and fitness of initial population of PSO, SPSO, and ASPSO algorithms for different scenarios. Refer to **Table 2** for specific location settings.

Table 2. Parameters

Scenario	UAV position	User Position
Scenario 1	(600,400,220)	(2100,2100,200)
Scenario 2	(2400,3500,190)	(3500,500,200)
Scenario 3	(1000,2000,150)	(3500,800,200)
Scenario 4	(3200,400,190)	(500,2300,200)

The population size is 1000, the particles number is 4,  $w_{\min} = 0.1$  and  $w_{\max} = 0.7$ ,  $\eta_1 = 1.3$  and  $\eta_2 = 1.5$ . Linearly decreasing inertia weight method is adopted. The climbing angle of SPSO is set as  $\psi \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right)$ . The **Table 3** compares the initialization times and initial fitness of different algorithms in the same scenario. The comparison situation as shown in Figure 5.

Table 3. Result

	1		2		3		4	
	T	F	T	F	T	F	T	F
PSO	17	17939	11	18485	10	26466	7	27820
SPSO	3	10818	50	15106	11	16436	1000	Inf
ASPSO	1	9809	9	12490	1	16805	4	18665

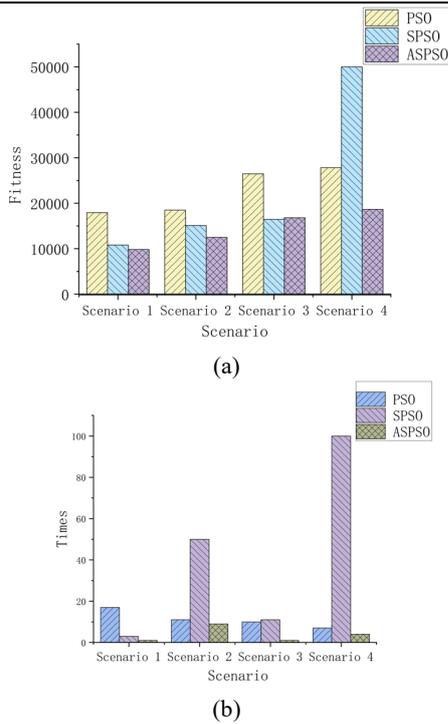


Figure 5. Comparison of Iteration Times and Fitness

From the above analysis, it can be seen that in any situation, ASPSO can find its initial solution with fewer iteration times. The search method can accelerate the efficiency of the algorithm. Although sometimes its initial fitness value is not the minimum value among the three comparison algorithms, it can also be minimized as much as possible through subsequent iterations. Although PSO can obtain initial solutions every time, its fitness value is relatively high and the quality of the solutions is relatively

low. SPSO was unable to initialize successfully in certain scenarios due to environmental constraints. The initialization strategy for complex environments has significant advantages in initialization speed and initial solution quality, providing a good foundation for the subsequent search of the algorithm.

### Analysis of Multi Strategy Optimization Method

To analyze the effectiveness of multi strategy optimization method, compare ASPSO, SPSO, and PSO. Set up different scenarios for simulation, and the setting of algorithm parameters is the same as the previous section. Visualize all paths and compare the fitness of each algorithm. Refer to **Table 4** for specific location settings.

Table 4. Position Parameters

Scenario	UAV position	User Position
Scenario 1	(600,1350,190)	(3500,2400,200)
Scenario 2	(500,2500,190)	(3500,500,200)

In Scenario 1, the path plots of each algorithm are compared with the fitness in Figure 6. The blue line is ASPSO algorithm, red line represents the SPSO algorithm, and the yellow line represents the PSO algorithm.

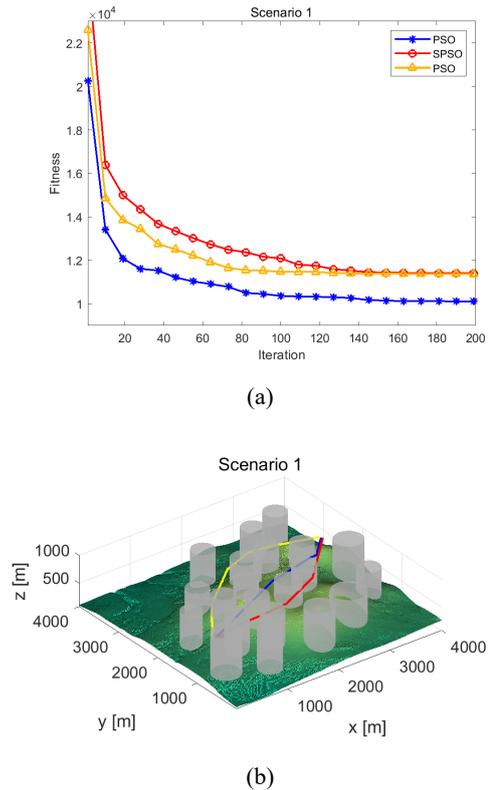
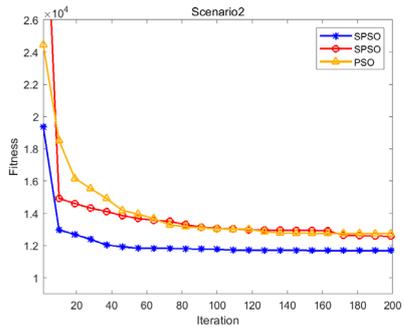
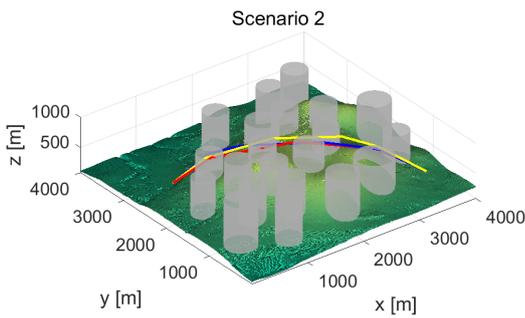


Figure 6. Scenario 1

In Scenario 2, the path plots of each algorithm are compared with the fitness as shown in Figure 7.



(a)



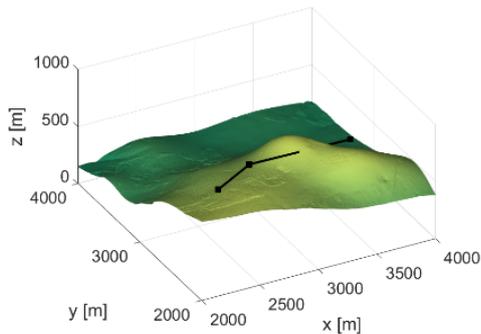
(b)

Figure 7. Scenario 2

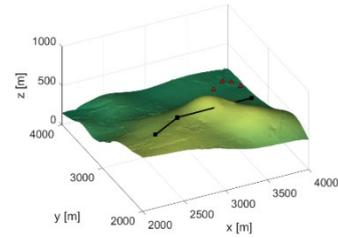
Analysis shows that regardless of whether the target point needs to pass through a complex environment, the ASPSO algorithm can always quickly obtain high quality solutions, with fast convergence speed and high solution quality. In the later stages of iteration, it can also continuously optimize and reduce fitness based on the better solution.

### Path Optimization Process

The process of path optimization strategy based on virtual control points is as follows. The initial path has a collision with the terrain. For each small path segment, a single dimensional small step search method is performed. For search points that do not meet the height constraint conditions, virtual points are constructed according to the terrain conditions, as Figure 8. Red triangle represents the constructed virtual points.



(a)



(b)

Figure 8. Initial Path and Virtual Control Points

For path segments with virtual control points, using the constructed virtual points as control points, keeping the starting and ending points of the original path segment unchanged, construct a Bezier curve for path smoothing, as shown in Figure 9. The optimized curve is shown by the yellow line.

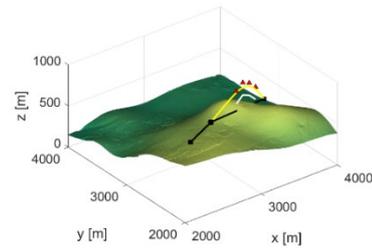
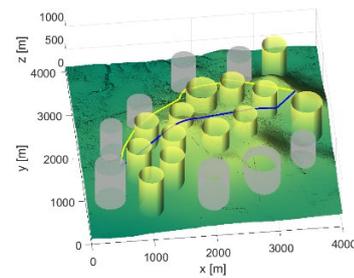
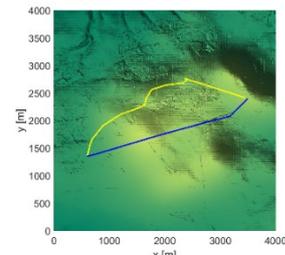


Figure 9. Optimized Path

If there are enough particles, collisions can also be avoided in situations with large terrain fluctuations. Set the algorithm particle count to 13, do not construct virtual control points, and compare it with the ASPSO algorithm, as Figure 10. The blue line is path produced by ASPSO. The yellow line is the comparison path with a higher number of particles.



(a)



(b)

Figure 10. Path Comparison Image

The comparison of fitness is shown in Figure 11. The comparison reveals that ASPSO still has a significant advantage over other methods with higher particle counts.

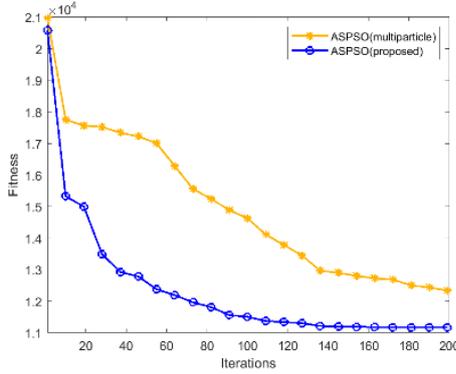


Figure 11. Fitness Comparison Image

## 4 Enhanced Collision Avoidance Vector Field

In this paper, the ECAVF algorithm is developed as an improvement over the original Collision Avoidance Vector Field algorithm [14]. The enhancements primarily focus on two aspects: the optimization of the vector field for moving obstacles and the synthesis of a multi-obstacle vector field.

### 4.1 Collision Avoidance Vector Field

To prevent collisions between the UAV and obstacles, the collision avoidance vector field  $f$  must satisfy the following non-penetration condition at any point  $p^b$  on the obstacle boundary.

$$f(p^b) \cdot h(p^b) \geq 0 \quad p^b \in \chi^b \quad (33)$$

Here,  $h(p^b)$  is the unit vector perpendicular to the obstacle boundary and pointing outward. The vector field  $f$  is defined as the superposition of the repulsive force  $f_r$  from the obstacle and the attractive force  $f_a$  toward the target position, as shown in Equation (34).

$$f(P) = f_r(P) + f_a(P) \quad (34)$$

The repulsive force of the obstacle is expressed as:

$$f_r(P) = \begin{cases} \gamma(P) R \|P_e - P\| h(P^C(P)) & \|P - P^C(P)\| \leq \rho \\ 0 & \|P - P^C(P)\| > \rho \end{cases} \quad (35)$$

where  $P_e$  is the target position of the UAV, and  $P^C(P) \in \chi^b$  represents the point on the obstacle boundary closest to point  $P$ .

The attractive force toward the target position is expressed as:

$$f_a(P) = \begin{cases} \|P_e - P\| R h(P_e - P) & \|P - P^C(P)\| \leq \rho \\ P_e - P & \|P - P^C(P)\| > \rho \end{cases} \quad (36)$$

Therefore, the force acting on the UAV at position  $P$  is expressed as:

$$f(P) = \begin{cases} \|P_e - P\| [\gamma(P) h(P^C(P)) + h(P_e - P)] & \|P - P^C(P)\| \leq \rho \\ P_e - P & \|P - P^C(P)\| > \rho \end{cases} \quad (37)$$

Where  $\gamma$  is a correlation factor determined by the relationship between the obstacle and the UAV's position, defined as follows:

$$\gamma(P) = \frac{1}{2} + \frac{\lambda d(P)}{\sqrt{1 + (2\lambda d(P))^2}} \quad (38)$$

Where  $d(P)$  is determined by the distance between the obstacle and the UAV, as well as the obstacle's influence distance, specifically expressed as follows:

$$d(P) = \frac{1}{\rho_1(P)} + \frac{1}{\rho_2(P)} \quad (39)$$

Where  $\rho_1(P)$  represents the closest distance between the UAV and the obstacle surface, denoted as  $\rho_1(P) = \|P - P^C(P)\|$ ,  $\rho_2(P)$  is  $\rho_2(P) = \rho_1(P) - \rho$ , and  $\lambda$  is a parameter determining the shape of the  $\gamma$  curve.

If the obstacle is in motion, its velocity is denoted as  $V_o$ , and the velocity of the obstacle boundary is also  $V_o$ . The movement of the obstacle causes changes in the vector field. Here, only the component of the obstacle's velocity in the normal direction of its boundary that is greater than zero is considered to influence the vector field ( $V_o \cdot h > 0$ ). This ensures that the UAV is not attracted to the obstacle when it is moving away. To account for the obstacle's motion, an additional term  $f_v(P) = \gamma(P) V_o$  is introduced into the vector field. The total vector field within the obstacle's influence range is expressed as:

$$f_{sum}(P) = f(P) + f_v(P) \quad (40)$$

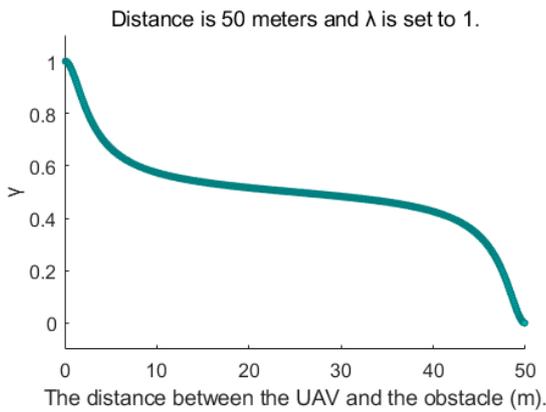
Considering the scenario with multiple obstacles, let  $d_j = \|P - P_j^C(P)\|$  be the distance from point  $P$  to the nearest point  $P_j^C(P)$  on the boundary of obstacle  $j$ . When there are  $N$  obstacles, the vector field is extended by weighting and summing the local CAVFs, with the weights determined by their distances to the UAV's current position. This ensures that the vector field at the boundary of each obstacle is solely determined by the local vector field  $h_k(P)$  associated with that obstacle:

$$h(P) = \sum_{k=1}^N w_k(P) h_k(P) \quad (41)$$

$$w_k(P) = \frac{\prod_{j \neq k} d_k}{\sum_{k=1}^N \prod_{j \neq k} d_j} \quad (42)$$

## 4.2 Optimization of the Vector Field for Moving Obstacles

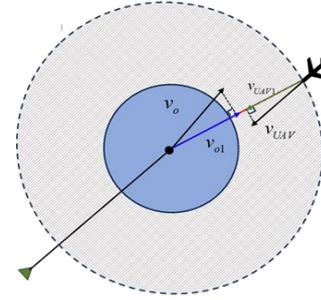
When moving obstacles are present in the scenario, the changes in the vector field are typically more complex. During operation, the UAV can detect environmental changes within a certain range, such as the position, size, and velocity of obstacles. The direction, speed, and position of the obstacles influence the vector field differently. In the presence of moving obstacles, the original CAVF algorithm, as referenced in Equation (40), multiplies the obstacle's velocity by  $\gamma$  as an increment. The value of  $\gamma$  is related to the distance between the UAV and the obstacle. As shown in Figure 12.



**Figure 12.** The graph of the  $\gamma$  function.

When the obstacle moves slowly and there is no significant opposing motion relative to the UAV, the original method can successfully avoid the obstacle. However, when the obstacle moves at a high speed or exhibits a pronounced opposing motion relative to the UAV, the repulsive force from the obstacle and the additional vector field generated by its motion are insufficient to enable the UAV to urgently avoid the obstacle in a dynamic state. As a result, the UAV may ultimately collide with the obstacle.

To address the obstacle avoidance problem for UAVs in complex dynamic environments, a dynamic vector field adjustment factor based on the relative velocity between the obstacle and the UAV is proposed. This adjustment factor dynamically adapts the avoidance strategy by incorporating relative velocity information to meet the requirements of obstacle avoidance under varying obstacle motion states. When the obstacle moves slowly and there is no significant opposing motion relative to the UAV, the original method can effectively avoid the obstacle. However, in scenarios with high relative velocities or rapidly moving obstacles, the avoidance strategy requires further optimization. The specific process is as follows.



**Figure 13.** Schematic diagram of the adjustment factor construction.

The unit vector from the UAV to the closest point on the obstacle is denoted as  $h$ . When the angle between the obstacle's velocity direction and the unit vector is acute, the obstacle tends to move toward the UAV. In this case, considering the obstacle's velocity as  $v_o$ , its velocity component along the line connecting the obstacle's centre and the UAV is  $v_{o1}$ . The UAV's velocity is  $v_{uav}$ , and its velocity component along the same line is  $v_{uav1}$ , with a magnitude of  $\|v_{uav1}\|$ . The relative velocity between them is  $v_r = v_{uav1} - v_{o1}$ , with a magnitude of  $\|v_r\|$ . The adjustment factor is defined as follows.

$$k = \frac{\|v_r\|}{\|v_{uav1}\|} \quad (43)$$

The vector field in the presence of moving obstacles is then defined as:

$$f_{sum}(P) = f(P) + k \times h \quad (44)$$

Within the influence range of the obstacle, when the relative velocity between the UAV and the obstacle is high, the construction of the vector field prioritizes obstacle avoidance as the dominant strategy. At this stage, the obstacle avoidance term dominates the vector field, ensuring that the UAV can promptly adjust its course to evade the obstacle. Once the obstacle is successfully avoided and no longer poses a threat, the vector field gradually transitions to guide the UAV back toward the target point, ensuring the continuity and completeness of the mission.

## 4.3 Construction Method of the Hybrid Vector Field

During the UAV motion planning process, it is essential to consider the combined effects of multiple obstacles on the UAV's velocity. When the UAV is within the influence range of multiple obstacles, its flight state is subject to various interferences, making the construction of a reasonable hybrid vector field crucial. The algorithm proposed in this section designs the synthesis of the vector field from multiple perspectives and dynamically adjusts

the vector field. Based on the direction and distance between the UAV and the target position, as well as the relative positions and velocities of the obstacles, the vector field weights are determined, enabling the UAV to avoid obstacles while efficiently moving toward the target position. Additionally, a buffer mechanism is introduced to smooth the velocity vector, reducing the impact of instantaneous fluctuations on the UAV's motion.

When multiple obstacles are present, the algorithm calculates the cosine of the angle between each obstacle and the UAV's position, combines it with the distance to the obstacle to determine the weight factor, and then computes the weighted potential field. The specific details are as follows:

For a given moment, the UAV's current position  $P$  is within the influence range of  $I$  obstacles. The center position of the  $i$ -th obstacle is  $P_{obs_i}$ , the current velocity is denoted as  $v_{obs_i}$ , and the vector from the  $i$ -th obstacle to the UAV position is given by:

$$\cos \theta_i = \frac{v_{obs_i} \cdot d_{P_{obs_i} \rightarrow P}}{\|v_{obs_i}\| \cdot \|d_{P_{obs_i} \rightarrow P}\|} \quad (45)$$

Here,  $\cos \theta$  reflects the influence of each obstacle's motion direction on the UAV. If this value is negative, it is replaced by an infinitesimal positive number. The weight calculation method based on the cosine of the angle is given by:

$$motionW_i = \frac{\prod_{j \neq i}^I \cos \theta_j}{\sum_{i=1}^I \prod_{j \neq i}^I \cos \theta_j} \quad (46)$$

Let  $dist_i$  denote the distance between the  $i$ -th obstacle and the UAV. Then the distance weighting is given by:

$$disW_i = \frac{\prod_{j \neq i}^I dist_j}{\sum_{i=1}^I \prod_{j \neq i}^I dist_j} \quad (47)$$

Therefore, the vector field weighting for the  $i$ -th obstacle is given by:

$$W_i = a_1 \cdot disW_i + a_2 \cdot motionW_i \quad (48)$$

Where  $a_1$  and  $a_2$  are constants, set to 0.9 and 0.1 respectively in this study. Let  $f_i$  denote the vector field of the  $i$ -th obstacle. Then the combined influence of  $I$  obstacles on the UAV is calculated as the linear weighted sum of their corresponding vector fields, given by:

$$f_{sum} = \sum_{i=1}^I W_i \cdot f_i \quad (49)$$

To reduce the impact of instantaneous velocity fluctuations on the UAV's motion and enhance the stability and reliability of the algorithm, a buffer update and smoothing mechanism is introduced. A buffer of size  $M$ , denoted as  $prev\_v$ , is initialized to store the UAV's historical velocity vectors. All elements in the buffer are initialized as zero vectors or the UAV's current velocity vector. Each

time a new velocity vector  $v$  is calculated, it is stored in the buffer. During each velocity update, valid velocity vectors are extracted from the buffer, and their average is computed as the final velocity vector. By averaging the historical velocity vectors, the impact of instantaneous fluctuations on the UAV's motion is minimized, improving the algorithm's stability.

Additionally, the UAV's flight process must account for its velocity constraints. Given the maximum speed limit  $v_{max}$  for the UAV, when the UAV is outside the influence range of any obstacle, the vector field is solely determined by the attractive force from the target position. To incorporate the UAV's speed constraints, the vector field is calculated as follows:

$$f(P) = \begin{cases} P_e - P & \|P - P^C(P)\| > \rho, \|P_e - P\| < D_{od} \\ \frac{P_e - P}{\|P_e - P\|} * V_{max} & \|P - P^C(P)\| > \rho, \|P_e - P\| > D_{od} \end{cases} \quad (50)$$

Here,  $D_{od}$  is the oscillation avoidance distance of the vector field. By setting different distance intervals to control the vector field's intensity, the UAV can maintain a stable and relatively high speed during long-distance flight. When approaching the target position, the vector field's intensity decreases, effectively preventing motion oscillations caused by excessive speed and inertia, which could otherwise lead to overshooting the target position and difficulty in achieving precise and stable arrival. This significantly improves positioning accuracy.

#### 4.4 Experimental Studies and Comparative Analysis

To systematically validate the effectiveness of the dynamic vector adjustment factor based on relative velocity and the construction of the hybrid vector field, the algorithm is tested in a three-dimensional dynamic environment to evaluate its adaptability across different scenarios. This section conducts simulations in a  $400 \times 400 \times 200$  (unit: meters) three-dimensional space, deploying spherical obstacles to simulate UAV path planning tasks in airspace with dense obstacles. A progressive scenario design is adopted, generating datasets in three stages: Scenario 1 (low complexity) contains 3-8 obstacles, Scenario 2 (medium complexity) expands to 9-20 obstacles, and Scenario 3 (high complexity) includes 21-30 obstacles. Each scenario strictly generates 100 independent samples, totalling 300 environmental configurations. The role of the relative velocity-based adjustment factor is verified.

Obstacle generation employs a hybrid motion model: static obstacles are modelled based on the start-end point connection line, forming static threats near the flight path. For moving obstacles, speeds are maintained between 10-20 m/s, with a 50% probability of moving along the path direction (same or opposite to the UAV) and a 50% probability of moving toward the path center, creating cross-interference.

For UAV parameter settings, the maximum flight speed is 30 m/s, equipped with an omnidirectional binocular vision system and a three-dimensional infrared sensor on the fuselage bottom, with a ranging scope of 0.4-22 m and a detection range of 0.4-200 meters. During dynamic obstacle avoidance, the UAV can detect obstacle distributions within a 200-meter range. In summary, the obstacle avoidance effectiveness of the proposed method is compared and analyzed across the three scenarios of varying complexity, with statistical results as follows:

Table 5. Statistical results of obstacle avoidance success rates in multi-obstacle scenarios.

	Scenario 1	Scenario 2	Scenario 3
Original Method	100%	89%	71%
Improved Moving Obstacle Vector Field	100%	100%	87%
Improved Moving Obstacle Vector Field + Enhanced Hybrid	100%	100%	93%

Analysis shows that in Scenario 1 with fewer obstacles, both the original algorithm and the proposed methods achieve successful obstacle avoidance. As the number of obstacles increases in Scenario 2, the original method fails in some cases, while the two improved methods maintain a 100% success rate. In the more complex Scenario 3, all methods experience some failures, but the proposed method achieves the highest success rate, demonstrating strong robustness even in highly complex environments. To visually illustrate the obstacle avoidance process, a set of data from Scenario 1 is selected, and the proposed method is applied for local path obstacle avoidance. The obstacle positions, velocity parameters, and a radius of 40 meters are shown in Table 6. The path obstacle avoidance process is depicted in Figure 14 and 15.

Table 6. Table of Obstacle Positions and Velocity Parameters.

Position (m)	Velocity (m/s)	Position (m)	Velocity (m/s)
(333,29,191)	(-13,11,-3)	(138,252,125)	(14,13,-14)
(358,65,57)	(0,0,0)	(141,233,99)	(15,17,0)
(13,409,103)	(-7,-6,12)	(141,183,66)	(0,0,0)
(62,253,114)	(9,8,-9.3)	(198,224,90)	(9.5,15,15)

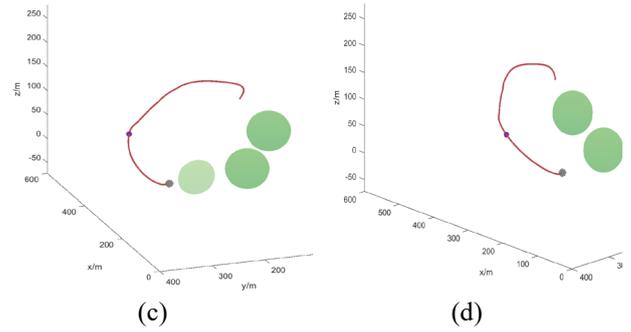
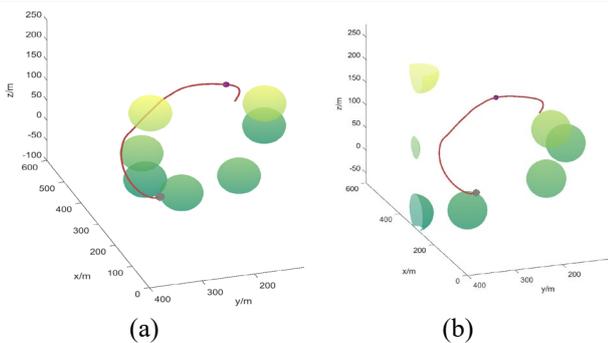


Figure 14. Obstacle Avoidance Process Using the Proposed Method in Scenario 1

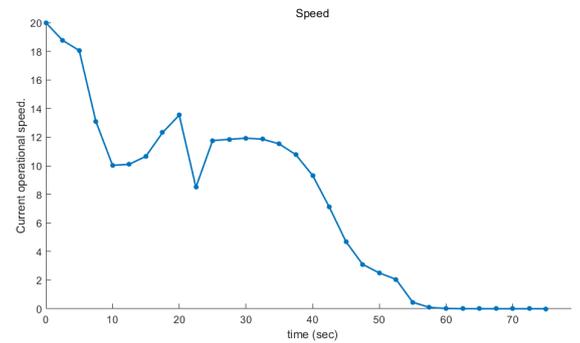


Figure 15. Velocity Curve During the Obstacle Avoidance Process.

## 5 Conclusions

We propose a ASPSO method based on SPSO. Based on the transmission model, determine the height between the drone and the user. The ASPSO algorithm first initializes based on environmental information, which can effectively shorten the initialization times and upgrade the initial solution for different scenarios. Introducing the idea of differential evolution during the search process makes it easy to escape form the local optima during the early stages, and can continue to explore in the neighbourhood space of high-quality solutions in the later stages of iteration. Finally, a path optimization strategy is implemented to avoid collisions with terrain and ensure flight safety. Compared with other algorithms, ASPSO algorithm has excellent performance, can obtain high quality solutions, and has better exploration ability in complex three-dimensional environments.

The proposed ECAVF algorithm addresses the issue of obstacle avoidance failure in scenarios with multiple moving obstacles by introducing a dynamic vector field adjustment factor based on relative velocity. When the UAV operates within the influence range of an obstacle, the vector field is dynamically adjusted according to the relative velocity between the UAV and the obstacle. When the relative velocity is high, the primary task is to avoid the

obstacle, after which the UAV proceeds toward the target position. During the synthesis of the vector field, the weights are determined by considering the direction and distance between the UAV and the target position, as well as the relative positions and velocities of the obstacles. This enhances obstacle avoidance capabilities and provides an effective guarantee for the safe flight of the UAV.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China Key Project (Grant No. 62431009). This work is an extended version of our conference paper “UAV Path Planning in Complex Environments for UAV Assisted Networks” previously presented at WiSATS 2024.

## References

- [1] Debnath D, Vanegas F, Sandino J, et al. A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications[J]. *Remote Sensing*, 2024, 16(21): 4019. [DOI:10.3390/rs16214019](https://doi.org/10.3390/rs16214019)
- [2] Luo J, Tian Y, Wang Z. Research on unmanned aerial vehicle path planning[J]. *Drones*, 2024, 8(2): 51. [DOI:10.3390/drones8020051](https://doi.org/10.3390/drones8020051)
- [3] Li Y, Wei W, Gao Y, et al. PQ-RRT\*: An improved path planning algorithm for mobile robots. *Expert systems with applications*. 2020; 152:113425. [DOI:10.1016/j.eswa.2020.113425](https://doi.org/10.1016/j.eswa.2020.113425)
- [4] Pehlivanoglu YV, Pehlivanoglu P. An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. *Appl Soft Computing*. 2021; 112:107796. [DOI:10.1016/j.asoc.2021.107796](https://doi.org/10.1016/j.asoc.2021.107796)
- [5] Zhao D, Cai G, Wang Y, et al. Path planning of obstacle-crossing robot based on golden sine grey wolf optimizer[J]. *Applied Sciences*, 2024, 14(3): 1129. [DOI:10.3390/app14031129](https://doi.org/10.3390/app14031129)
- [6] Hu G, Huang F, Shu B, et al. MAHACO: Multi-algorithm hybrid ant colony optimizer for 3D path planning of a group of UAVs[J]. *Information Sciences*, 2025, 694: 121714. [DOI:10.1016/j.ins.2024.121714](https://doi.org/10.1016/j.ins.2024.121714)
- [7] Song B, Wang Z, Zou L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Appl Soft Computing*. 2021; 100:106960. [DOI:10.1016/j.asoc.2020.106960](https://doi.org/10.1016/j.asoc.2020.106960)
- [8] Sonny A, Yeduri SR, Cenkeramaddi LR. Autonomous UAV path planning using modified PSO for UAV-assisted wireless networks. *IEEE Access*. 2023. [DOI:10.1109/ACCESS.2023.3293203](https://doi.org/10.1109/ACCESS.2023.3293203)
- [9] Liu X, Liu Y, Chen Y, et al. Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(8): 7957-7969. [DOI:10.1109/TVT.2019.2920284](https://doi.org/10.1109/TVT.2019.2920284)
- [10] Yu Z, Si Z, Li X, et al. A novel hybrid particle swarm optimization algorithm for path planning of UAVs. *IEEE Internet Things J*. 2022;9(22):22547-22558. [DOI:10.1109/JIOT.2022.3182798](https://doi.org/10.1109/JIOT.2022.3182798)
- [11] Liu X, Yu Y, Li F, et al. Throughput maximization for RIS-UAV relaying communications. *IEEE Intelligent Transportation Systems*. 2022;23(10):19569-19574. [DOI:10.1109/TITS.2022.3161698](https://doi.org/10.1109/TITS.2022.3161698)
- [12] Mozaffari M, Saad W, Bennis M, et al. Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization. *IEEE Trans Wireless communication*. 2017;16(12):8052-8066. [DOI:10.1109/TWC.2017.2756644](https://doi.org/10.1109/TWC.2017.2756644)
- [13] Phung MD, Ha QP. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl Soft Computing*. 2021; 107:107376. [DOI:10.1016/j.asoc.2021.107376](https://doi.org/10.1016/j.asoc.2021.107376)
- [14] Braquet M, Bakolas E. Vector field-based collision avoidance for moving obstacles with time-varying elliptical shape. *IFAC-Papers Online*. 2022;55(37):587-592. [DOI:10.1016/j.ifacol.2022.11.246](https://doi.org/10.1016/j.ifacol.2022.11.246)