UAV Path Planning in Complex Environments for UAV Assisted Networks

Xinyue Chang, Liang Ye(✉), Lin Ma, and Shuyi Chen

Harbin Institute of Technology, Harbin, China
yeliang@hit.edu.cn

Abstract. This paper considers a network scenario assisted by unmanned aerial vehicles (UAVs). In a complex environment with dense obstacles, the UAVs are deployed to provide downlink services to users. Users are located in areas with dense obstacles. Based on the transmission model, the target position of path is determined, this method is used to seek optimal path under constraint conditions. This work proposes the ASPSO (Adaptive spherical vector-based Particle Swarm Optimization) algorithm. Firstly, an initialization way is designed. The method sets different parameters for scene requirements, effectively shortens the initial times. Secondly, differential evolution is introduced during the search process, and a multi-strategy optimization method is proposed. Increasing the search space in the early stages of iteration is beneficial for get rid of the local good solution. In the late stages of iteration, small disturbance is introduced to continue exploring in the neighborhood space of high-quality solution. Finally, a way for path improvement with virtual control points is proposed to smooth the trajectory and reduce fitness. In this paper, the method is compared with PSO (Particle Swarm Optimization) and SPSO (Spherical Vector-based PSO), the ASPSO can quickly obtain high quality initial solutions and has better exploration ability in complex three dimensional environments.

Keywords: UAV Assisted Networks · PSO · Path Planning

1 Introduction

UAVs have strong maneuverability [1], and with the advancement of technology, they are used in detection, and search and rescue. UAVs path planning need to consider task time, flight distance, and safety, and UAVs may encounter many obstacles during flight, such as buildings, mountains, radar areas, etc. The path of UAVs is research hotspot, and it is necessary to design the path under various constraint conditions, fulfill task [2].

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, 6], particle swarm optimization (PSO) [7] and other ways.

H.-H. Chen and W. Meng (Eds.): WiSATS 2024, LNICST 606, pp. 230–244, 2025.
https://doi.org/10.1007/978-3-031-86203-8_19

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 rely 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 water 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 a function.

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

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 [2]. The coordinates of the circle center in the horizontal plane of cylinder i is $(x_i, y_i, 0)$. Figure 1 is the environment.

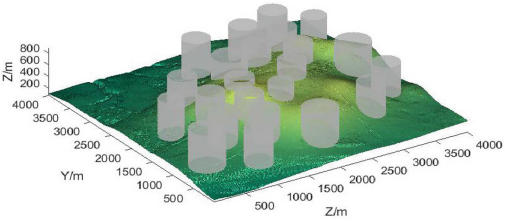

Fig. 1. Flight Environment

The communication link from UAV to user can be considered as an air to ground transmission. 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 [12]:

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

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

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

where $K_0 = (4\pi f_c/c)^2$, defined α as path loss parament. The f_c is carrier frequency and c is light speed.

r(t) is the user's rate at time t

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

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 trajectory, a UAV flight trajectory T, C represents the set of n necessary trajectory 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

$$F_1(T) = \sum_{i=1}^{n-1} \left\| \overrightarrow{C_i C_{i+1}} \right\| \tag{5}$$

UAVs are also constrained in terms of flight height, with minimum and maximum height limits between the flight 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 $a_{best} = \frac{a_{\min}+a_{\max}}{2}$, the altitude cost is computed as

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

Summing A_i for all points gives the height cost:

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

Assuming that all obstacle sets are O. The center 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 Fig. 2. The half of the central angle corresponding to the secant of circle C is denoted as θ_u, and θ_u is calculated as

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

To avoid collisions, the maximum value of θ_u is θ_m, and θ_m is calculated as

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

The ratio of the two is recorded as θ_r, and $\theta_r = \frac{\theta_u}{\theta_m}$.
The threat cost calculation for obstacles is as follows (Fig. 3):

$$\begin{cases} F_3(T) = \sum_{i=1}^{n-1} B\left(\overrightarrow{C_i C_{i+1}}\right) \\ B\left(\overrightarrow{C_i C_{i+1}}\right) = \begin{cases} \infty & d_o < R_o + l \\ e^{-\frac{\theta_r}{\theta_r - 1}} & R_o + l < d_o < R_o + l + Z \\ 0 & d_o > R_o + l + Z \end{cases} \end{cases} \tag{10}$$

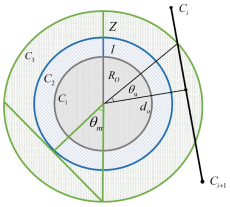

Fig. 2. Obstacle Threat

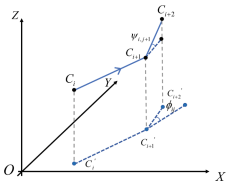

Fig. 3. Path smoothness.

The smoothness of the path is also an important metric to look at. The direction angle ϕ_i and the climbing angle ψ_i. they are calculated as

$$\phi_{i,i+2} = \arctan\left(\frac{\left\|\overrightarrow{C_i' C_{i+1}'} \times \overrightarrow{C_{i+1}' C_{i+2}'}\right\|}{\overrightarrow{C_i' C_{i+1}'} \cdot \overrightarrow{C_{i+1}' C_{i+2}'}}\right) \tag{11}$$

$$\psi_i = \arctan\left(\frac{z_{i+1} - z_i}{\left\|\overrightarrow{C_i' C_{i+1}'}\right\|}\right) \tag{12}$$

The path smooth cost is then computed as:

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

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

$$F_{\text{sum}} = \text{w}_1 F_1(T) + \text{w}_2 F_2(T) + \text{w}_3 F_3(T) + \text{w}_4 F_4(T) \tag{14}$$

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 trajectory 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:

$$P = (r_1, \phi_1, \psi_1, r_2, \phi_2, \psi_2, ..., r_n, \phi_n, \psi_n) \tag{15}$$

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

$$\Delta p = \left(r_1', \phi_1', \psi_1', r_2', \phi_2', \psi_2', ..., r_n', \phi_n', \psi_n'\right) \tag{16}$$

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

$$v_{ij}^{t+1} = wv_{ij}^{t} + \eta_1\gamma_1\left(p_{best}^{t} - u_{ij}^{t}\right) + \eta_2\gamma_2\left(g_{best}^{t} - u_{ij}^{t}\right) \tag{17}$$

$$u_{ij}^{t+1} = u_{ij}^{t} + v_{ij}^{t+1} \tag{18}$$

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 ϕ_i and the climbing angle ψ_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:

$$\psi = \operatorname{atan}\left(\frac{2 \cdot \mathrm{h}}{\mathrm{r}_{\max}}\right) \tag{19}$$

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 trajectory. 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 Fig. 4.

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, refer to the selection rate function to determine the size, as shown below:

$$r_s = \left(r_f - r_e\right) \cdot \left(\left(\frac{i}{Inter}\right)^2 - \frac{2i}{Inter}\right) + r_s \tag{20}$$

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

$$s\_num = npop \cdot r_s \tag{21}$$

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

$$phi_p = \left(phi_{p,1}, phi_{p,2}, \ldots, phi_{p,N-1}, phi_{p,N}\right) \tag{22}$$

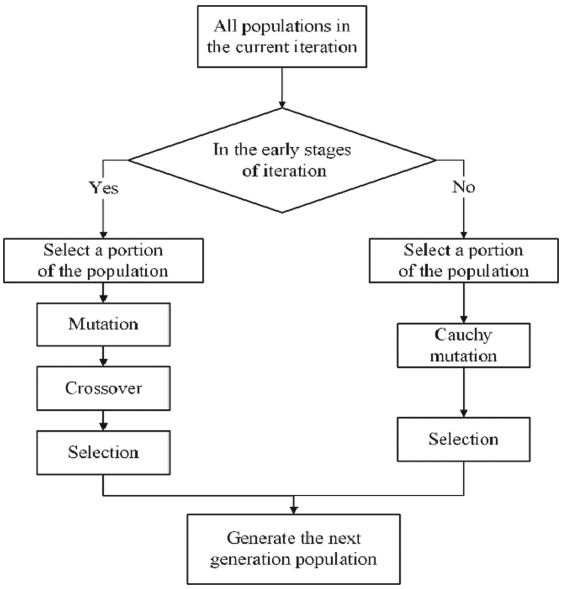

Fig. 4. Multi strategy Flowchart

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

$$v_p = phi_p + F \cdot (phi_{r1} - phi_{r2}) \tag{23}$$

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

$$u_{p,n}=\begin{cases} v_{p,n} & rand < CR \ or \ n = j \\ phi_{p,n} & otherwise \end{cases} \quad n = 1, 2, \cdots, N \tag{24}$$

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

$$phi_{p,n} = phi_{p,n} + k \cdot C\left(R_p; x_0, \gamma\right) \cdot phi_{p,n} \tag{25}$$

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 Trajectory 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 trajectory, there may be collisions between the trajectory and the ground, which directly poses a safety threat to the flight nodes. Therefore, we propose a trajectory optimization strategy based on virtual control points. On the basis of the initial trajectory, a single dimensional small step search method is used to find trajectory 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 trajectory while ensuring low fitness, reducing trajectory 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.

3.4.1 Single Dimensional Small Step Search Method

In practical environments, UAVs move in three dimensional space, and the initial trajectory is a three dimensional trajectory. Here, a single dimensional small step search method is proposed. For the initial trajectory, the initial trajectory is divided into small trajectory 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 trajectory is defined as L_{xyz}, the starting point of the trajectory is (X_S, Y_S, Z_S) and the endpoint is (X_E, Y_E, Z_E). The points for the entire trajectory is $\{X_S, Y_S, Z_S; X_1, Y_1, Z_1; ...; X_n, Y_n, Z_n; X_E, Y_E, Z_E\}$. The trajectory L_{xyz} is actually composed of $L_{xyz,i}$, $i = 1, 2, \cdots, n, n+1$.

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

$$\begin{cases} \Delta x_1 = X_1 - X_S \\ \Delta x_2 = X_2 - X_1 \\ \quad ... \\ \Delta x_n = X_n - X_{n-1} \\ \Delta x_{n+1} = X_{\mathrm{E}} - X_n \end{cases} \tag{26}$$

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

$$z_{iv} = Peak(x_{iv}, y_{iv}) + \alpha \cdot a_{best} \tag{27}$$

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

3.4.2 Bezier Fitting Based on Virtual Control Points.

For each small trajectory $L_{xyz,i}$, 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

$$B(t) = \sum_{i=1}^{n} B_{in}(t) P_i \tag{28}$$

The Bernstein basis function as follow

$$B_{in}(t) = C_n^i t^k (1-t)^{n-i}, i = 1, 2, ..., n \tag{29}$$

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 trajectory smoothing. The curve consists of five points. In this paper, if there are too many trajectory nodes, the higher height nodes are prioritized as control points. If there are less than five trajectory nodes, use a low order Bessel curve for optimization.

4 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 km $\times$ 4 km $\times$ 1 km. For the transmission model, the following Table 1 gives the parameter settings [9].

The P_{LoS} is 0.891 and the ending point is set directly above the user, with the UAV positioned at a height of 200 m from the user.

Table 1. Parameters Setting

Parameter	Description	Value
f_c	Carrier frequency	2 GHz
p	UAV output power	30 dBm
N_0	Noise power spectral	-70 dBm/Hz
B	Bandwidth	1 MHz
μ_{LoS}	Los additional path loss	3 dB
μ_{NLoS}	NLoS additional path loss	23 dB
b_1	Environmental parameters	0.36
b_2	Environmental parameters	0.21

4.1 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 Fig. 5.

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.

Table 3. Result

	Scen 1		Scen 2		Scen 3		Scen 4	
Algorithm	Time	Fitness	Time	Fitness	Time	Fitness	Time	Fitness
PSO	17	17939	11	18485	10	26466	7	27820
SPSO	3	10818	50	15106	11	16436	1000	999999
ASPSO	1	9809	9	12490	1	16805	4	18665

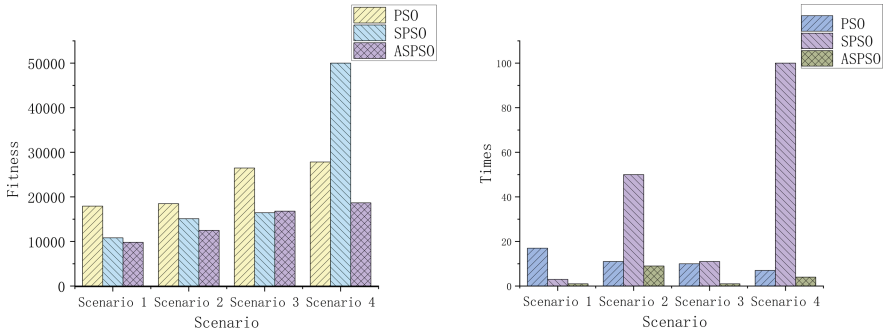

Fig. 5. Comparison of Iteration Times and Initial Fitness

4.2 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 trajectory plots of each algorithm are compared with the fitness in Fig. 6. The blue line is ASPSO algorithm, red line represents the SPSO algorithm, and the yellow line represents the PSO algorithm.

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

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

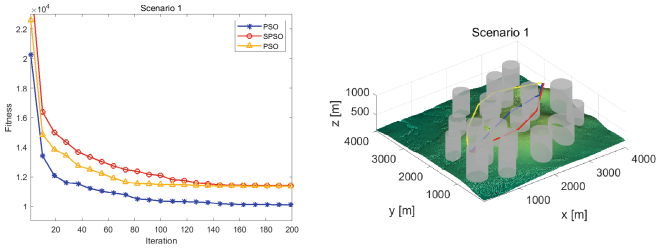

Fig. 6. Scenario 1

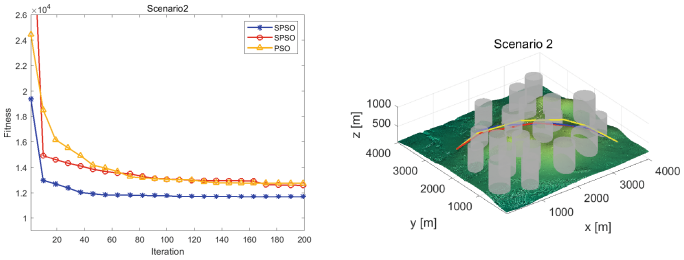

Fig. 7. Scenario 2

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.

4.3 Trajectory Optimization Process

The process of trajectory optimization strategy based on virtual control points is as follows. The initial trajectory has a collision with the terrain. For each small trajectory 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 Fig. 8. Red triangle represents the constructed virtual points.

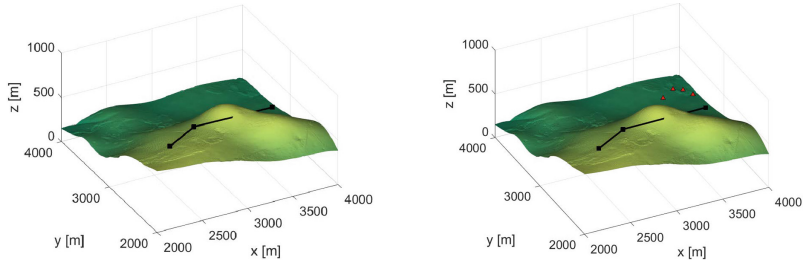

Fig. 8. Initial Trajectory and Virtual Control Points

For trajectory segments with virtual control points, using the constructed virtual points as control points, keeping the starting and ending points of the original trajectory

segment unchanged, construct a Bezier curve for trajectory smoothing, as shown in Fig. 9. The optimized curve is shown by the yellow line.

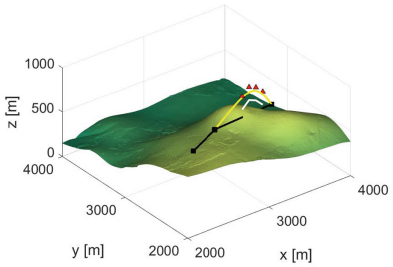

Fig. 9. Optimized Trajectory

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 Fig. 10. The blue line is trajectory produced by ASPSO. The yellow line is the comparison trajectory with a higher number of particles.

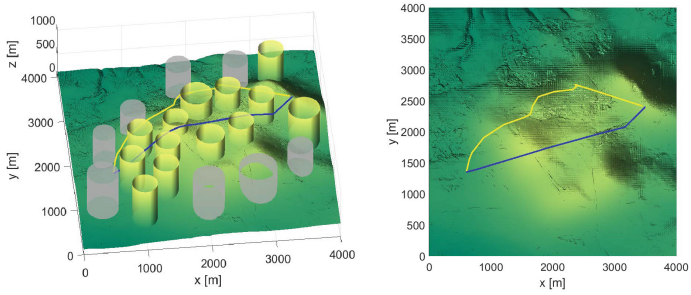

Fig. 10. Trajectory Comparison Image

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

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 neighborhood space

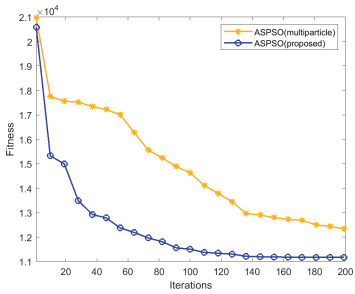

Fig. 11. Fitness Comparison Image

of high quality solutions in the later stages of iteration. Finally, a trajectory 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.

Acknowledgement. This paper is supported by Key Research and Development Plan of Heilongjiang Province, Grant No. JD22A001.

References

1. Liu, X., Lai, B., Lin, B., et al.: Joint communication and trajectory optimization for multi-UAV enabled mobile internet of vehicles. IEEE Trans. Intell. Transp. Syst. **23**(9), 15354–15366 (2022)
2. Chen, Y., Yu, J., Mei, Y., et al.: Modified central force optimization (MCFO) algorithm for 3D UAV path planning. Neurocomputing **171**, 878–888 (2016)
3. Li, Y., Wei, W., Gao, Y., et al.: PQ-RRT*: an improved path planning algorithm for mobile robots. Expert Syst. Appl. **152**, 113425 (2020)
4. Pehlivanoglu, Y.V., Pehlivanoglu, P.: An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. Appl. Soft Comput. **112**, 107796 (2021)
5. Chen, X., Kong, Y., Fang, X., et al.: A fast two-stage ACO algorithm for robotic path planning. Neural Comput. Appl. **22**, 313–319 (2013)
6. Dewangan, R.K., Shukla, A., Godfrey, W.W.: Three dimensional path planning using Grey wolf optimizer for UAVs. Appl. Intell. **49**, 2201–2217 (2019)
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 Comput. **100**, 106960 (2021)
8. Sonny, A., Yeduri, S.R., Cenkeramaddi, L.R.: Autonomous UAV path planning using modified PSO for UAV-assisted wireless networks. IEEE Access **11**, 70353–70367 (2023)
9. Liu, X., Liu, Y., Chen, Y., et al.: Trajectory design and power control for multi-UAV assisted wireless networks: a machine learning approach. IEEE Trans. Veh. Technol. **68**(8), 7957–7969 (2019)
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. **9**(22), 22547–22558 (2022)
11. Liu, X., Yu, Y., Li, F., et al.: Throughput maximization for RIS-UAV relaying communications. IEEE Trans. Intell. Transp. Syst. **23**(10), 19569–19574 (2022)

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 Commun. **16**(12), 8052–8066 (2017)
13. Phung, M.D., Ha, Q.P.: Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. Appl. Soft Comput. **107**, 107376 (2021)