IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 March 2024, accepted 27 March 2024, date of publication 4 April 2024, date of current version 16 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3384976

RESEARCH ARTICLE

An Improved Plant Growth Algorithm for UAV Three-Dimensional Path Planning

HENG XIAO[1], ZHENJIE MU[1], WEN ZHOU[2], AND HUI ZHANG[1]
[1]Faculty of Civil Aviation and Aeronautics, Kunming University of Science and Technology, Kunming 650500, China
[2]College of Systems Engineering, National University of Defense Technology, Changsha 4100733, China

Corresponding author: Zhenjie Mu (20212245030@stu.kust.edu.cn)

This work was supported in part by Yunnan Province Science and Technology Department under Grant 202201AT070080.

ABSTRACT In response to the challenge of three-dimensional path planning for unmanned aerial vehicles (UAVs), this paper introduces a Multi-Strategy Plant Growth Path Planning (MSPGPP) algorithm. Initially, a threat sphere and obstacle avoidance cone model are constructed to ensure safety in avoidance and enhance algorithm accuracy and convergence velocity, while B-spline curves are introduced to improve the quality of the path. Next, the concept of threat level is incorporated along with UAV maneuverability to establish a multi-objective optimization function, enabling the algorithm to plan optimal paths. Finally, an algorithm framework is developed, and comparative simulation results of the improved algorithm against the original algorithm and other optimization algorithms are presented in a three-dimensional environment. The results demonstrate that compared to the original algorithm, the improved algorithm better balances obstacle avoidance safety and UAV maneuverability, planning safer and more effective paths. Additionally, it achieves higher accuracy and faster convergence. Compared to other optimization algorithms, the improved algorithm has better optimization capabilities, higher computational efficiency, and superior path quality.

INDEX TERMS Unmanned aerial vehicle, path planning, multi-objective optimization, plant growth, obstacle avoidance cone.

I. INTRODUCTION

The application field of UAVs is becoming increasingly broad. In the military domain, UAVs contribute to reconnaissance, target tracking, and coordinated strikes, significantly enhancing the efficiency and success rate of military operations [1], [2], [3]. In civil domains, UAVs are used for infrastructure inspection, logistics delivery, agricultural spraying, search and rescue tasks, showing great application value and market potential [4], [5], [6], [7]. The challenges in these application domains can be summarized as three-dimensional path planning problems for UAVs [8], which are global optimization problems under multiple constraints, considering environmental characteristics, mission requirements, and UAV performance constraints.

Traditional path planning algorithms such as the A* algorithm [9], Dijkstra's method [10], Rapidly expanding random tree (RRT) algorithm [11], and artificial potential field method [12] have matured significantly but are mostly used in two-dimensional scenarios and prone to issues like unreachable targets and local minima. In recent years, many intelligent optimization algorithms inspired by biological evolution have been successfully applied to UAV path planning, such as particle swarm optimization [13], genetic algorithms [14], ant colony algorithms [15], and newer algorithms like dung beetle algorithm [16], grey wolf algorithm [17], and artificial bee colony algorithm [18].

The main challenges in UAV three-dimensional path planning lie in modeling optimization and algorithm improvement. The former requires consideration of the rationality and applicability of path planning modeling and the safety of UAV flight. The latter demands excellent global optimization capabilities, high computational efficiency, and accuracy in complex three-dimensional environments. In terms of modeling optimization, literature [19] have established UAV path planning models based on terrain and turning constraints, and combined with fuel consumption and threat to establish objective functions, utilizing multi-modal

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

multi-objective evolutionary algorithms to provide various effective decisions; and literature [20] has segmented sub-paths according to the number of hazard sources and the distance to them, calculating threat costs accordingly. However, these modeling processes often use the distance between the UAV and obstacles as the sole criterion, leading to either excessively large avoidance distances or reduced safety. Hence, different avoidance distances and their impact on path scenarios should be considered. Additionally, the actual flight of UAVs should comprehensively consider UAV maneuverability to ensure safer and more practical flight paths. In terms of algorithm improvement, chaos sequences and various adaptive strategies have been employed to enhance the search efficiency and precision of PSO algorithms [21], improving the quality of solutions for UAV three-dimensional path planning problems. The proposed Harris Hawks algorithm introduces adaptive chaos and core population dynamic partition strategies [22], enhancing the algorithm's late-stage search capabilities, combined with adaptive dynamic cloud optimal solution perturbation strategies, to increase the ability to escape local optima. These improved algorithms contribute to better path planning quality, but due to the large search space and high computational efficiency requirements of the UAV three-dimensional path planning problem, they are limited by the characteristics of the algorithms themselves. A comprehensive UAV three-dimensional path planning solution based on bionic principles proposes a new Plant Growth Path Planning (PGPP) algorithm [23], which is adaptable, stable, and efficient. However, it lacks global optimization in path planning, resulting in higher path costs. The original paper also points out the potential application value of the PGPP algorithm in the field of optimization problems.

This paper focuses on modeling optimization, introducing the concept of threat level combined with maneuverability and path smoothness, considering UAV maneuverability. In algorithm improvement, it combines the advantages of the PGPP algorithm with various models and strategies [24], [25], [26], [27] to propose a new MSPGPP algorithm, validated through simulation in the established UAV three-dimensional path planning model. The experimental results show that the improved algorithm plans more cost-effective and globally optimal UAV flight paths while maintaining the original algorithm's rapid and accurate pathfinding capabilities.

The main contributions of this paper are summarized as follows:

- Modeling the UAV three-dimensional path planning problem as a multi-objective optimization problem, aiming to enhance the capability of the original PGPP algorithm in solving for optimal paths and reducing path costs.
- Proposing a multi-strategy improved MSPGPP scheme for optimal UAV path planning. The proposed scheme integrates threat spheres, obstacle avoidance cones, B-spline curves, and multi-objective optimization to enhance the algorithm's optimization performance and improve path quality.
- Implementing the proposed MSPGPP algorithm, the original PGPP algorithm, and three famous optimization algorithms including PSO, RRT, and GWO. Simulation experiments validate the effectiveness of the proposed model and strategies.
- Assessing the efficacy of the MSPGPP algorithm in UAV path planning, comparing it with the original PGPP and other optimization algorithms, verifying the effectiveness and practicality of the proposed algorithm.

II. UAV THREE-DIMENSIONAL PATH PLANNING

This section describes the problem of three-dimensional path planning for Unmanned Aerial Vehicles (UAVs). It first presents the kinematic model of the UAV, then transforms the path planning problem into a multi-objective optimization problem through modeling.

A. PROBLEM DEFINITION

UAV path planning refers to generating an optimal flight path within a known flight area that can go from a mission's starting point to a target point, fulfilling various path planning requirements. In this paper, the UAV flight path X_i is represented as a collection of n waypoints, with each waypoint corresponding to coordinates in three-dimensional space $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$.

B. UAV KINEMATIC MODEL

This paper assumes a safe radius R_u for the UAV in three-dimensional space. Considering the shape and motion uncertainty of obstacles, the obstacle is simplified as a spherical model. O represents the center of the obstacle, and R_o represents the threat radius of the obstacle. Taking a certain moment as an example, the positions of the UAV and the obstacle are P_u and P_o, respectively; the velocity vector of the UAV is $\mathbf{V_u}$. Thus, the motion model of the UAV can be represented by equations (1)-(2).

$$P_u = (x_u, y_u, z_u) \tag{1}$$

$$\mathbf{V_u} = \begin{bmatrix} v_{ux} \\ v_{uy} \\ v_{uz} \end{bmatrix} = \begin{bmatrix} v_u \cos\theta_u \sin\varphi_u \\ v_u \cos\theta_u \cos\varphi_u \\ v_u \sin\theta_u \end{bmatrix} \tag{2}$$

where v_u represents the magnitude of the UAV's velocity vector, φ_u and θ_u the angle of the direction of the UAV's velocity vector, where φ_u is the yaw angle and θ_u is the pitch angle.

C. MULTI-OBJECTIVE OPTIMIZATION

This paper transforms the UAV three-dimensional path planning problem into a multi-objective optimization problem and establishes a multi-objective optimization function associated with constraints like path length, flight altitude, obstacle avoidance safety, UAV maneuverability, and path

smoothness. Then, it evaluates the quality of the UAV flight path X_i as a feasible solution to the multi-objective optimization function to find the path with the lowest comprehensive cost as the optimal solution.

1) PATH LENGTH

The cost associated with path length Z_1 can be calculated by equation (3), representing the distance between two adjacent nodes of the path X_i as $\left\|\overrightarrow{P_{ij}P_{i,j+1}}\right\|$.

$$Z_1(X_i) = \sum_{j=1}^{n-1} \left\|\overrightarrow{P_{ij}P_{i,j+1}}\right\| \tag{3}$$

2) FLIGHT ALTITUDE

When UAVs perform low-altitude tasks, flight altitude needs to be considered. First, the UAV path nodes are controlled within a safe altitude range, then the height difference between two adjacent nodes is calculated using a two-point formula to avoid abrupt increases or decreases in flight altitude. The cost of height difference is given by equation (4), where h_{min} and h_{max} represent the minimum and maximum flight altitudes of the UAV, and h_k represents the unit cost of height difference. The total cost function related to flight altitude is shown in equation (5).

$$H_{ij} = \begin{cases} h_k \left| z_{i,j+1} - z_{ij} \right|, & if\, h_{\min} \le z_{ij} \le h_{\max} \\ \infty, & otherwise \end{cases} \tag{4}$$

$$Z_2(X_i) = \sum_{j=1}^{n-1} H_{ij} \tag{5}$$

3) THREAT LEVEL

Based on the concept of safe avoidance, this paper introduces the concept of threat level, considering different distance scenarios to establish a dangerous area for avoidance. The threat level includes the cost of threat distance and threat time, meaning the smaller the distance between the UAV and the threat, the greater the threat level; the longer the flight time in the danger zone, the greater the threat level.

The costs of flight time and threat distance are given by equations (6) and (7), respectively.

$$C_t(\overrightarrow{P_{ij}P_{i,j+1}}) = \frac{\left\|\overrightarrow{d_1 d_n}\right\|}{v_u} \tag{6}$$

$$\begin{cases} C_d(\overrightarrow{P_{ij}P_{i,j+1}}) = \sum_{k=1}^{n} D_k \\ D_k = \begin{cases} L - l_k, & if\, R_o \le l_k \le L \\ \infty, & if\, l_k \le R_o \end{cases} \end{cases} \tag{7}$$

where $C_t(\overrightarrow{P_{ij}P_{i,j+1}})$ represents the path flight time cost function; $C_d(\overrightarrow{P_{ij}P_{i,j+1}})$ represents the path threat distance cost function.

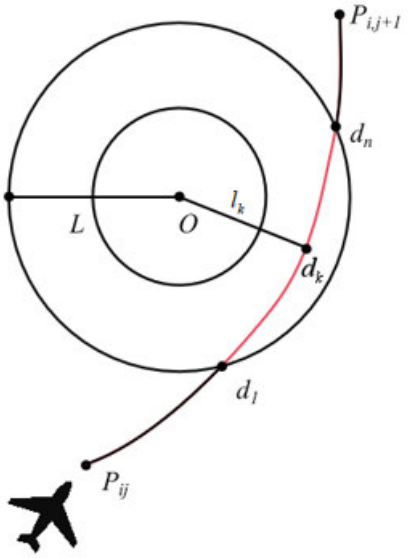

FIGURE 1. Threat Level cost. As shown in Fig. 1, the radius of the hazardous area is L (which is jointly determined by the safety radius of the UAV and the radius of the obstacle), and the distance l_k between d_k and O can be determined according to the Euclidean distance formula, and the subpath $\overrightarrow{d_1 d_n}$ consisting of all the nodes d_k for which l_k is less than L can be further determined.

The sum of threat time and threat distance costs for all segments yields the threat level cost function:

$$Z_3(X_i) = \sum_{j=1}^{n-1} C_t(\overrightarrow{P_{ij}P_{i,j+1}}) + C_d(\overrightarrow{P_{ij}P_{i,j+1}}) \tag{8}$$

4) MANEUVERABILITY

The path also needs to consider guiding the UAV through the obstacles in the avoidance range with a qualified and safe maneuver. Taking the point P_{ij} as an example, the vector $\overrightarrow{P_{i,j-1}P_{ij}}$ and the projection of $\overrightarrow{P_{i,j-1}P_{ij}}$ in the horizontal direction can be denoted as $\mathbf{k}_i = (x_{ij} - x_{i,j-1}, y_{ij} - y_{i,j-1})$ and $\mathbf{k}_{i+1} = (x_{i,j+1} - x_{ij}, y_{i,j+1} - y_{ij})$, respectively, as shown in equation (9), from which the yaw angle φ_{ij} and pitch angle θ_{ij} can be calculated.

$$\begin{cases} \varphi_{ij} = \arccos\left(\dfrac{\mathbf{k}_i^{\mathrm{T}} \mathbf{k}_{i+1}}{|\mathbf{k}_i| \cdot |\mathbf{k}_{i+1}|}\right) \\ \theta_{ij} = \arctan\left(\dfrac{\left|z_{i,j+1} - z_{ij}\right|}{\sqrt{(x_{i,j+1} - x_{ij})^2} + \sqrt{(y_{i,j+1} - y_{ij})^2}}\right) \end{cases} \tag{9}$$

The UAV's maximum yaw angle is set as φ_{max}, and the maximum pitch angle as θ_{max}. The maneuverability cost is directly proportional to φ_{ij} and θ_{ij} and inversely proportional to φ_{max} and θ_{max}. Thus, the turning angle cost function is shown in equation (10).

$$C_a(\overrightarrow{P_{ij}P_{i,j+1}}) = K_t \frac{\varphi_{ij}}{\varphi_{max}} + K_c \frac{\theta_{ij}}{\theta_{max}} \tag{10}$$

where $C_a(\overrightarrow{P_{ij}P_{i,j+1}})$ represents the path turning angle cost function; K_t and K_c are the cost coefficients for the yaw angle and pitch angle, respectively.

The sum of all nodes yields the maneuverability cost function:

$$Z_4(X_i) = \sum_{j=1}^{n-1} C_a(\overrightarrow{P_{ij}P_{i,j+1}}) \tag{11}$$

5) MULTI-OBJECTIVE OPTIMIZATION FUNCTION

Considering the costs related to path length, flight altitude, threat level, maneuverability associated with path X_i, the total cost function can be defined in the following form:

$$Z(X_i) = \sum_{k=1}^{4} b_k Z_k(X_i) \tag{12}$$

where Z_1 to Z_4 are the target cost functions associated with path length, flight altitude, threat level, maneuverability, respectively; b_k are weight coefficients; X_i is the decision variable.

With these definitions, the multi-objective optimization function $Z(X_i)$ is fully determined and can be used as input in the UAV path planning process.

III. MSPGPP ALGORITHM

This section describes the MSPGPP algorithm. It first introduces the principles and workflow of the original PGPP algorithm, followed by various improvement models and strategies for the original algorithm proposed in this paper.

A. ORIGINAL PGPP ALGORITHM

The Plant Growth Path Planning (PGPP) algorithm, inspired by the phototropism of plants and their ability to grow towards light while avoiding shadowy areas, reflects a similar characteristic in UAVs–navigating towards a target while avoiding obstacles. As a biomimetic algorithm, The primary feature of the PGPP algorithm is its inheritance of an adaptive ability of the plant to its environment. When applied to UAVs, this enables quick and efficient responses to obstacles. Additionally, as an incremental rapid search algorithm, PGPP boasts high computational efficiency and low complexity, making it suitable for application in optimization fields. This is particularly relevant for UAV path planning, where optimal solutions are sought using algorithms with these characteristics.

The PGPP algorithm simulates plant growth mechanisms to achieve UAV path planning. The algorithm mimics various characteristics of plant growth, using the apical bud or current node as the basic growth unit to calculate and find the UAV flight path. The growth of the apical bud in the PGPP algorithm mainly follows phototropism principles and negative geotropism, branching rules, and other plant growth mechanisms. Phototropism and negative geotropism are the main features of the PGPP algorithm.

As an example, taking the starting point $Start = (x_S, y_S, z_S)$ and the target point $Target = (x_T, y_T, z_T)$, with the obstacle as $P_o = (x_o, y_o, z_o)$, and the distance between the path node and the obstacle as l_k, the basic steps of the algorithm are as follows:

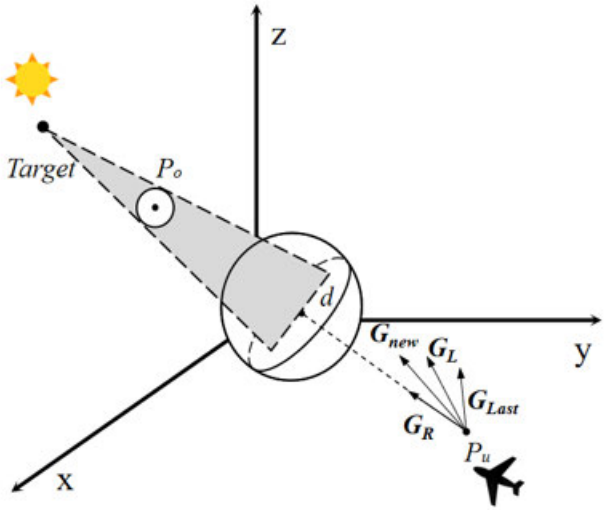

FIGURE 2. Growth principle. As shown in Figure 2, phototropism refers to the light-oriented growth behavior of plants, described as vector $\mathbf{G_L}$. *Target* represents the light source, i.e., the target point. The magnitude of $\mathbf{G_L}$ is directly proportional to the distance between the bud point and the target point. Negative geotropism refers to the characteristic of plants growing away from the ground against gravity, described as a force $\mathbf{G_R}$ always pointing towards the target point. This force helps prevent the bud from falling into local minima when approaching the target point due to a small $\mathbf{G_L}$. The growth vector $\mathbf{G_{new}}$ for the current cycle, guiding the growth of the apical bud to generate the UAV path, can be obtained by summing $\mathbf{G_L}$, $\mathbf{G_R}$, and the growth vector $\mathbf{G_{Last}}$ from the previous cycle.

Step 1: Initialization. Initialize variables such as start and target positions, read map data, set the starting point *Start* as the initial growth point $bud = (x_b, y_b, z_b)$, and proceed to step 2.

Step 2: Light intensity calculation. Calculate the light intensity of all buds in the search range, and the light intensity of the range covered by shadows will be weakened. As shown in Fig. 3, the search range is updated with a radius of d. The search range can be divided into several cube grids $\Omega_{i,j,k}$ with a side length of $bond$. Taking the point bud as the center of the coordinate axis, the coordinate point (x_g, y_g, z_g) in the search range satisfies equation (13).

$$\begin{aligned}
&(x_g, y_g, z_g) \\
&\quad \in \Omega_{i,j,k} \\
&\Omega_{i,j,k} = \begin{cases} i \cdot d \le x_g \le (i+1) \cdot bond \\ j \cdot d \le y_g \le (j+1) \cdot bond \\ k \cdot d \le z_g \le (k+1) \cdot bond \end{cases} \\
&\begin{cases} i \in \left[-floor(\frac{d}{bond} + 0.5), floor(\frac{d}{bond} + 0.5) - 1 \right] \\ j \in \left[-floor(\frac{d}{bond} + 0.5), floor(\frac{d}{bond} + 0.5) - 1 \right] \\ k \in \left[-floor(\frac{d}{bond} + 0.5), floor(\frac{d}{bond} + 0.5) - 1 \right] \end{cases}
\end{aligned} \tag{13}$$

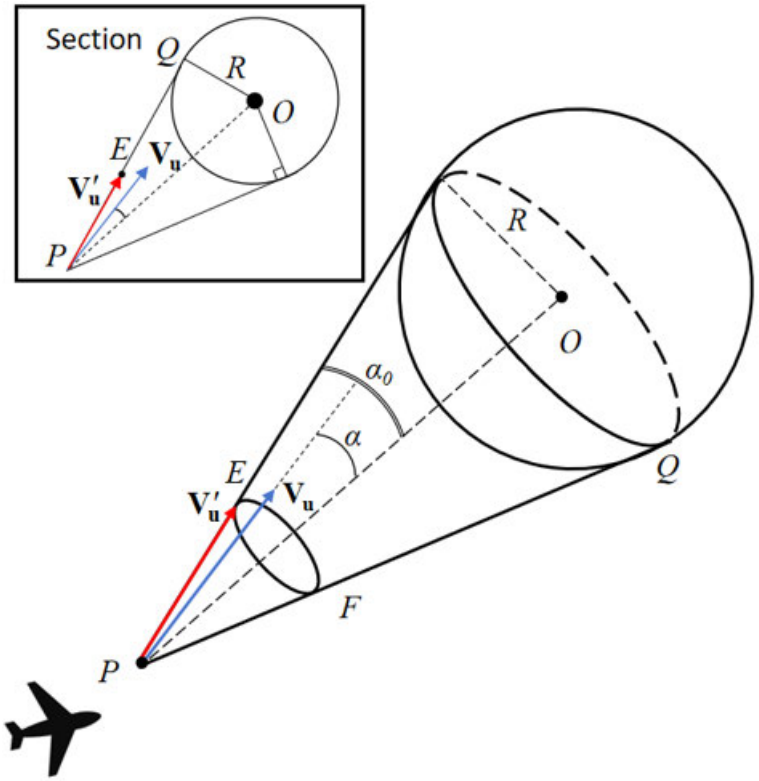

FIGURE 3. Obstacle Avoidance Cone. As shown in Figure 3, combining the threat sphere model, taking point P_u as the current position of the UAV, the straight line L_{uo} between P_u and P_o can be determined using the Euclidean distance formula. At this time, the length of L_{uo} is less than or equal to L_{safe}. Drawing a tangent line through point P_u that is tangent to the threat sphere model at point Q, the same applies to the other side, obtaining the obstacle avoidance cone model considering the adaptive safe distance.

Then, the total light intensity $Lsumi, j, k$ of the grid can be calculated using equations (14) and (15), and proceed to step 3.

$$L_i(x,y,z) = \begin{cases} \dfrac{K_L\left[(x_T-x_b)^2+(y_T-y_b)^2+(z_T-z_b)^2\right]}{(x-x_b)^2+(y-y_b)^2+(z-z_b)^2}, \\ \quad if\, l_k > R_o \\ 0, \quad otherwise \end{cases} \tag{14}$$

$$Lsum_{i,j,k} = \sum L_i(x_g, y_g, z_g) \tag{15}$$

Step 3: Random Branching. First, select several buds with the largest light intensity. According to the number of growth times, determine whether each bud meets the branching condition. The age of the bud will be updated after each growth, and exceeding the limit will turn it into a dead bud to prevent falling into a local unsolvable state. If the branching condition is met, determine the probability P_{branch} of generating new buds using equation (16).

$$P_{branch} = \frac{Lsum_{i,j,k}}{\sum Lsum_{i,j,k}} \tag{16}$$

where $\sum Lsum_{i,j,k}$ represents the sum of the light intensities of all the grids in the search range, and proceed to step 4.

Step 4: Calculation of Growth Vector. Represent the center of grid $\Omega_{i,j,k}$ as $P_{center} = (x_c, y_c, z_c)$. Calculate the growth vector for all buds, and combine it with the growth vector from the previous cycle with equal weights to form a new growth vector $\mathbf{G}_{new}$. The magnitude of the growth vector is obtained using equation (17). Then, determine the endpoint $[x_{grow}, y_{grow}, z_{grow}]$ of the growth vector $\mathbf{G}_{new}$, which can determine the growth direction of the bud, i.e., the UAV's yaw angle φ_u and pitch angle θ_u, and proceed to step 5.

$$\begin{cases} G_L = (x_c - x_b)i + (y_c - y_b)j + (z_c - z_b)k \\ G_R = (x_T - x_b)i + (y_T - y_b)j + (z_T - z_b)k \\ G_{new} = K_1 \cdot G_L + K_2 \cdot G_R + K_3 \cdot G_{Last} \end{cases} \tag{17}$$

Step 5: Plant Growth. Each bud grows according to the branching growth vector. Determine whether each bud reaches the endpoint during growth. If it reaches the endpoint, the search ends. Execute step 6. If not, return to step 2 until a bud reaches the endpoint.

Step 6: Obtain the UAV Flight Path. At the end of the search, determine the path according to the parent node of each path point.

Algorithm 1: The pseudo-code of the MSPGPP algorithm

Require: *Start* , *Target* *Map* , *path* , P_o , R_o , *MaxIt*
Ensure: *Bestpath* , *Bestcost*

1 **Threat_Ball** （ R_o ） Update threat sphere radius R ;
Start store to *path* ;
2 **for** i = 1,2,..., *MaxIt*
MSPGRP algorithm path-finding:
3 **While! (Rearch_Target(** *path* **)) do**
Light_Cal （ d ） Calculate Light Intensity $Lsum_{i,j,k}$
Random_Branch （ $Lsum_{i,j,k}$ ） Branch probability P_{branch}
4 **Growth_Cal** （ P_{branch} ） Calculate the growth vector $\mathbf{G}_{new}$
Plant_Growth （ $\mathbf{G}_{new}$ ） The new point *Bud* store to *path*
5 **End while**
6 **Path_Opt** （ *path* ） B-spline curve optimization *newpath*
Solve the multi-objective optimization function:
Path_cost （ *newpath* ） Calculate the total cost *pathcost* ;
if *pathcost* < *Bestcost*
7 *Bestpath* = *newpath* Update the optimal path;
Bestcost = *pathcost* Update the optimal cost;
End if
8 **End for**

FIGURE 4. The pseudocode of the MSPGPP algorithm. The pseudocode of the MSPGPP algorithm workflow is shown in Figure 4, where 1-6 belongs to the model and strategy improvement part, and 7 belongs to the multi-objective optimization function part.

B. THREAT SPHERE MODEL

The original PGPP algorithm does not consider the safe range of obstacle avoidance during path planning, resulting in a short processing time for UAV obstacle avoidance and prone to collisions. Therefore, this paper expands the initial obstacle information and constructs an obstacle threat sphere model based on the obstacle radius, UAV velocity, and safe radius, ensuring a safe distance and sufficient time for the UAV to avoid obstacles.

As calculated and analyzed by the model, as shown in equation (18), taking O as the center of the threat sphere and r as the expanded radius, a threat sphere model with radius R is determined.

$$\begin{cases} r = 2\left\|\mathbf{V_u}\right\| \dfrac{|R_o - R_u|}{R_o + R_u} \\ R = R_u + r \end{cases} \tag{18}$$

C. OBSTACLE AVOIDANCE CONE

When planning paths, UAVs aim to minimize the number of calculations and the time taken by the algorithm. In step 3 of the original PGPP algorithm, a random growth strategy is used for pathfinding, leading to excessive randomness in the growth vector calculated in step 4. This may result in too many calculations to obtain the optimal path, which is not conducive to the accuracy and timeliness of the algorithm. Therefore, this paper introduces the obstacle avoidance cone model to improve the branching growth strategy, as follows:

First, to ensure the safety and accuracy of obstacle avoidance and improve the utilization rate of the original path to reduce flight costs, define the UAV's adaptive safe distance L_{safe} with obstacles, as shown in equation (19).

$$L_{safe} = 3\left(R_o + \left\|\mathbf{V_u}\right\| \frac{|R_o - R_u|}{R_o + R_u}\right) \tag{19}$$

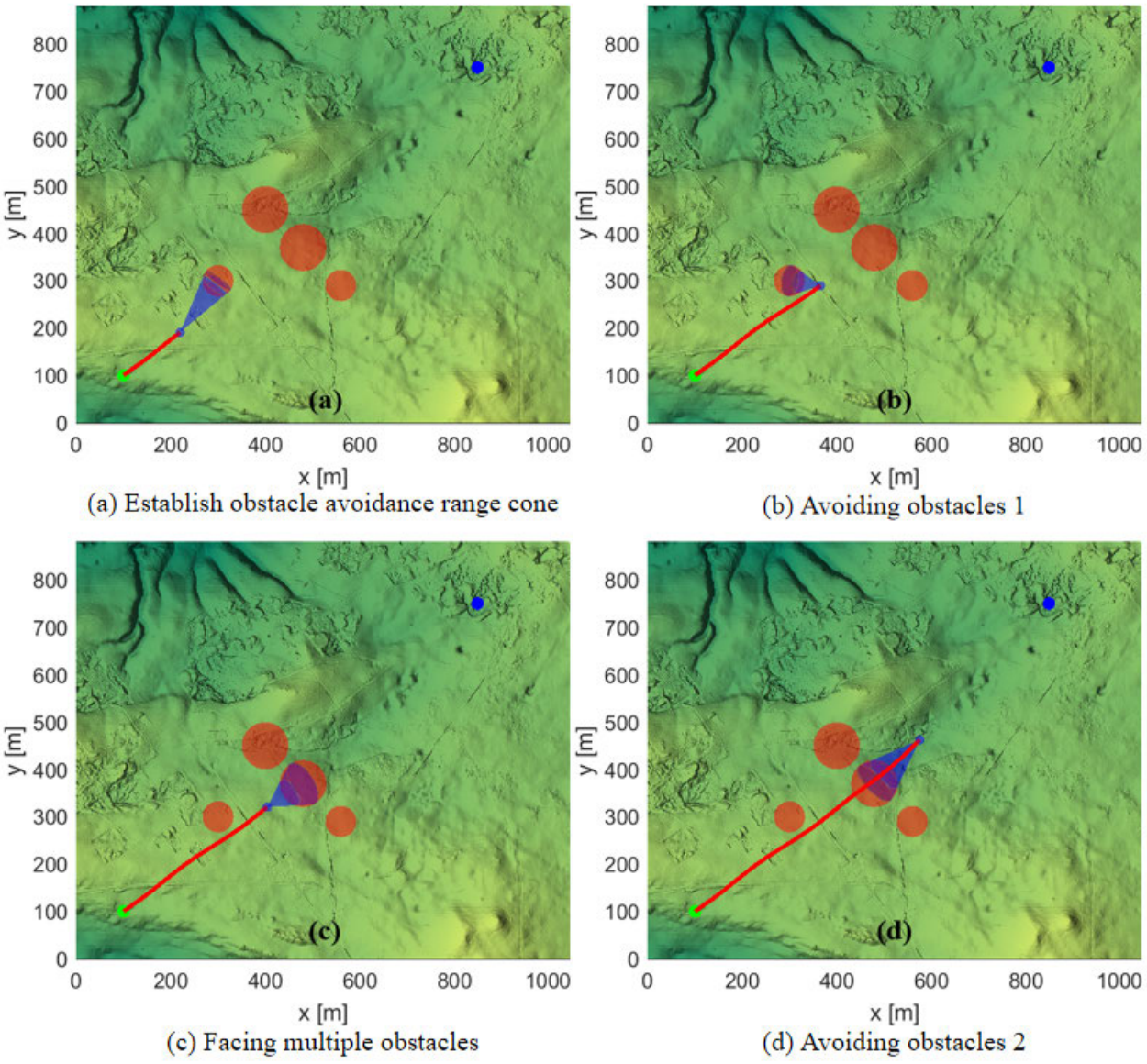

FIGURE 5. The Process of Model and Strategy Analysis Experiment. As can be seen from Figures (a) and (b), when the distance between the UAV and the obstacle is less than Lsafe, the obstacle avoidance cone model is effective and guides the UAV to safely avoid obstacles. Figure (c) shows that when the UAV faces multiple obstacles, the model prioritizes the closest obstacle. Figures (b) and (d) jointly indicate that the UAV maintains a safe distance to avoid obstacles.

As analyzed by the algorithm simulation results, the numerical setting of L_{safe} is related to the UAV's velocity, safe radius, and the threat radius of obstacles. L_{safe} should not be too small to ensure that the UAV can safely avoid obstacles while meeting its maneuverability conditions. L_{safe} should not be too large to avoid the UAV prematurely leaving the original path, leading to conservative obstacle avoidance issues.

In Figure 3, the angle α between the UAV's velocity $\mathbf{V_u}$ and L_{uo}, and the angle α_0 between L_{uo} and L_{PE} can be calculated using equations (20) and (21).

$$\sin\alpha_0 = \frac{R}{\|L_{uo}\|} \tag{20}$$

$$\cos\alpha = \frac{L_{uo}\mathbf{V_u}}{\|L_{uo}\|\,\|\mathbf{V_u}\|} \tag{21}$$

This model is only effective when the distance between the UAV and the obstacle is less than the adaptive safe distance. When the UAV needs to avoid multiple obstacles, the obstacle avoidance cone model prioritizes the closest obstacle.

When avoiding obstacles, the UAV needs to change its velocity vector. As shown in Figure 4, the current velocity vector of the UAV is $\mathbf{V_u}$. To achieve obstacle avoidance, the UAV needs to deviate from the obstacle avoidance cone model, i.e., fly with the desired velocity vector $\mathbf{V_u'}$ to achieve obstacle avoidance. Points E and F are the critical points of the UAV's desired velocity vector for obstacle avoidance.

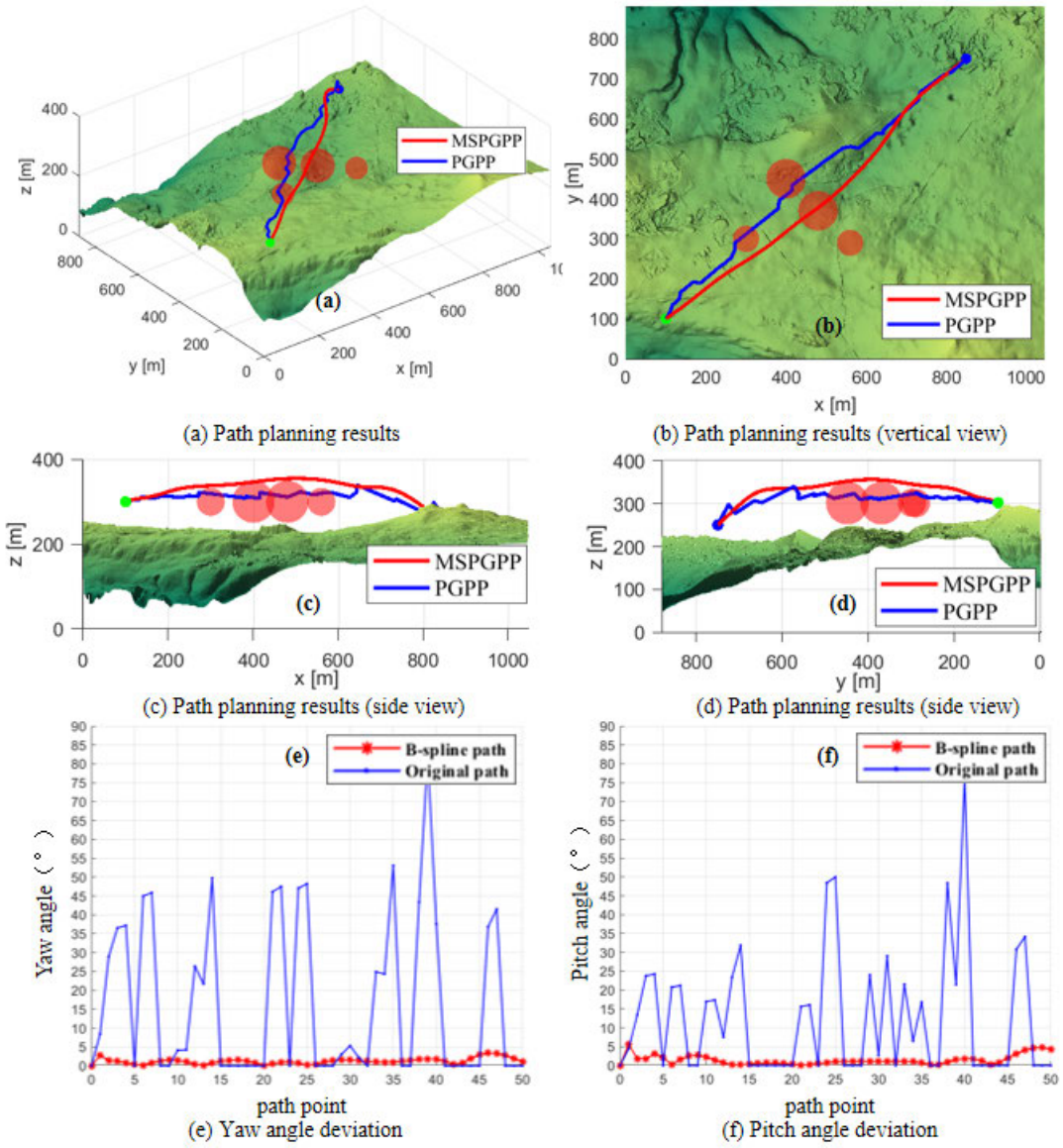

FIGURE 6. Results of experiments on model and strategy analysis. As can be seen from Figures 6 (a) - (d), the path planned by the original algorithm passes through the threat sphere, posing a collision risk, while the improved algorithm always effectively avoids the threat sphere and maintains a safe distance from obstacles, providing higher safety. In addition, the path planned by the improved algorithm is smoother than that of the original algorithm; from Figure (e) and (f), it can be seen that the path planned by the improved algorithm has smaller yaw and pitch angle deviations, while the original algorithm has more deflections and larger deviations, resulting in higher path costs.

Taking the solution of the critical point E of the desired velocity as an example, set the current coordinates of the UAV as (x_u, y_u, z_u), the current velocity vector $\mathbf{V_u} = [v_{ux}, v_{uy}, v_{uz}]^T$, and the coordinates of the center O point of the obstacle as (x_o, y_o, z_o); set the unit vector of L_{PE} as $\mathbf{m} = [x, y, z]$. The unit vector $\mathbf{m}$ of L_{PE} can be determined using equations (22) to (24).

$$\sqrt{x^2 + y^2 + z^2} = 1 \tag{22}$$

$$\cos\alpha_0 = \frac{x(x_o - x_u) + y(y_o - y_u) + z(z_o - z_u)}{\sqrt{(x_o - x_u)^2 + (y_o - y_u)^2 + (z_o - z_u)^2}} \tag{23}$$

$$\cos(\alpha_0 - \alpha) = \frac{xv_{ux} + yv_{uy} + zv_{uz}}{\|\mathbf{V_u}\|} \tag{24}$$

Then, the desired velocity vector $\mathbf{V'_u}$ can be obtained using equation (27).

$$\mathbf{V'_u} = \left[v'_{ux}, v'_{uy}, v'_{uz}\right]^{\mathrm{T}} = (xv_{ux} + yv_{uy} + zv_{uz})\mathbf{m} \tag{25}$$

As shown in equations (26) and (27), by solving the desired velocity vector $\mathbf{V'_u}$, the directional angles φ'_u and θ'_u of the desired velocity vector can be obtained.

$$\cos\varphi'_u = \frac{v'_{uy}}{\sqrt{\left(v'_{ux}\right)^2 + \left(v'_{uy}\right)^2}} \tag{26}$$

$$\cos\theta'_u = \frac{v'_{uz}}{\sqrt{\left(v'_{ux}\right)^2 + \left(v'_{uy}\right)^2 + \left(v'_{uz}\right)^2}} \tag{27}$$

Finally, the growth direction of the bud that satisfies equation (28) can be determined, discarding growth vectors that do not meet the directional angle requirements.

$$\begin{cases} \varphi_u \geq \varphi'_u \\ \theta_u \geq \theta'_u \end{cases} \tag{28}$$

D. B-SPLINE OPTIMIZATION

The original paths generated by the PGPP algorithm consist of a series of path nodes, which are relatively scattered and have excessive deflections, leading to sudden large changes in direction during UAV flight. This may not meet the UAV's maneuverability constraints. Therefore, considering the UAV's yaw and pitch angle constraints, the original path needs to be optimized.

This paper uses B-spline curves to smooth the original path. B-spline curves are an effective method for constructing curves and surfaces, suitable for UAV path planning problems. The optimized path is smoother and more stable, meeting the UAV's maneuverability constraints and enabling effective flight mission execution. As shown in equation (29).

$$\begin{cases} P(u) = \sum_{i=0}^{n} B_i^n(u) P_i, \\ \qquad u \in \left\{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, 1\right\} \\ B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}, \\ \qquad i \in \{0, 1, \ldots, n\} \end{cases} \tag{29}$$

where n represents the order of the B-spline curve; u is the position parameter; P_i represents the control point; $P(u)$ is the new set of path nodes.

IV. SIMULATION AND VALIDATION

The experimental part was conducted in a computer testing environment with a Win10 (64-bit) system, Intel Core i7-7700 CPU, 2.8GHz, and 16GB RAM, using Matlab R2021a software. This section first compares the MSPGPP algorithm with the original PGPP algorithm, including experimental analyses of the obstacle avoidance cone model and path optimization strategy, as well as the multi-objective optimization function, to validate the effectiveness and efficiency of the model and strategies. Then, the improved algorithm is compared with other intelligent optimization algorithms under the same conditions in UAV three-dimensional path planning simulation experiments to verify the effectiveness of the improved algorithm in path planning problems.

A. COMPARISON WITH THE ORIGINAL ALGORITHM

This section compares the MSPGPP algorithm with the PGPP algorithm in path planning simulation experiments.

1) ANALYSIS OF IMPROVED MODELS AND STRATEGIES

To validate the effectiveness and applicability of the model and strategies, the MSPGPP algorithm, which only introduces the obstacle avoidance cone model and path optimization strategy, is compared with the original algorithm.

Digital Elevation Model (DEM) terrain data from the ALOS PALSAR database was used to construct a

TABLE 1. Model and strategy analysis experimental parameters.

Experimental parameters	Set value
MAP	[1045,879,400]
Start	(100,100,300)
Target	(850,750,250)
v_u	$20m/s$
R_u	$10m$
P_o	(400,450,300)
	(480,370,300)
	(560,290,300)
	(300,300,300)
R_o	$20m$
	$20m$
	$30m$
	$30m$

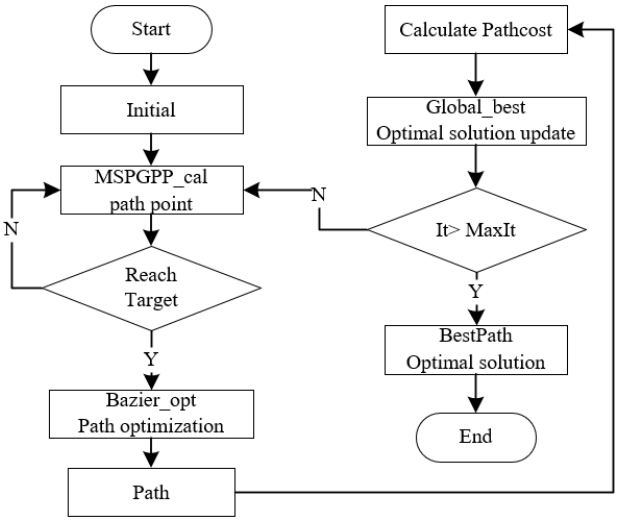

FIGURE 7. Steps of MSPGPP to solve the optimal path. As shown in Figure 7, after extracting its own and obstacle information, the UAV solves path nodes through the MSPGPP algorithm. Specific steps include light intensity calculation, establishing an obstacle avoidance cone model to determine the branching direction, calculating growth vectors, etc. If the target point is reached, the path is optimized using B-spline curves; otherwise, path nodes continue to be solved. After obtaining the path, it is stored and the total path cost is calculated. In each calculation, only the solution with the lowest comprehensive cost is retained. Until the iteration limit is reached, the optimal solution at this time is output as the optimal path.

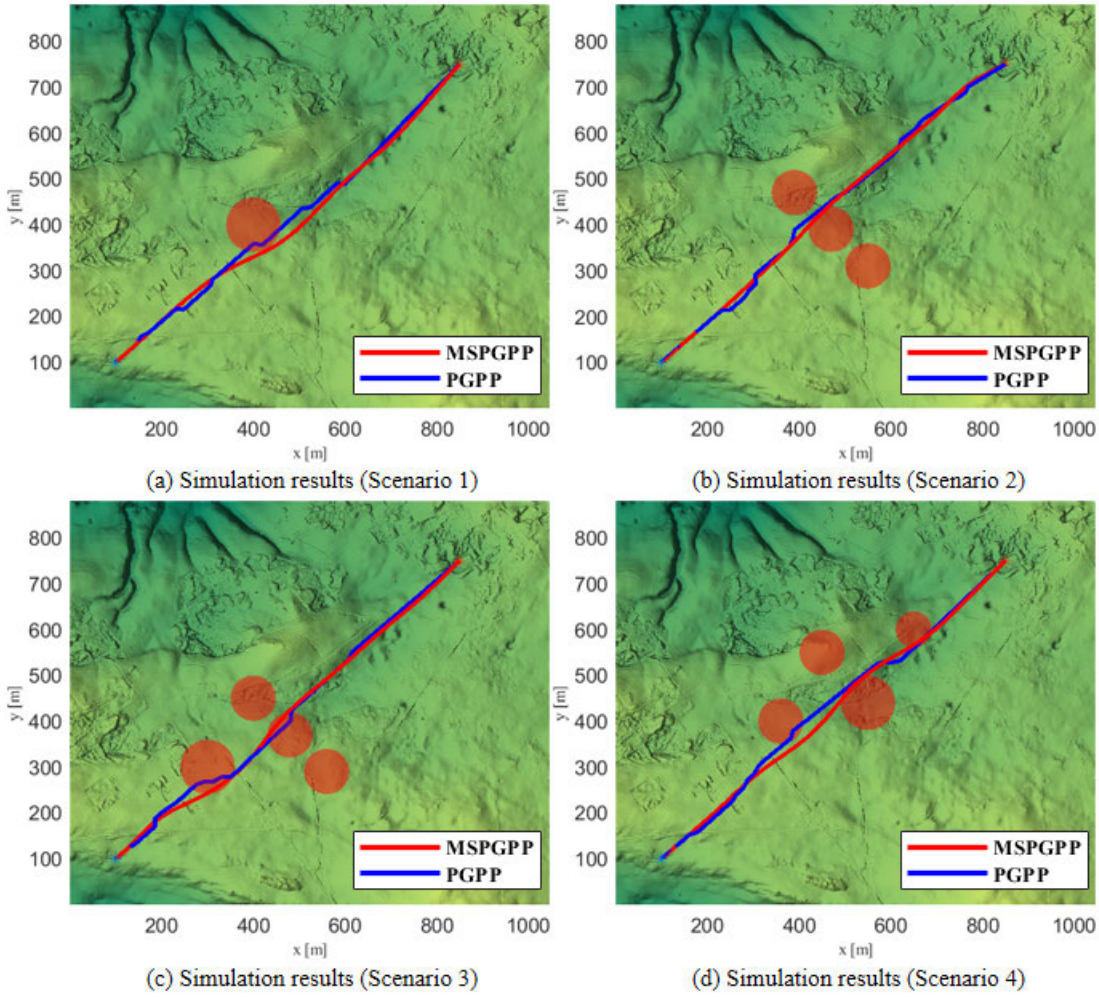

FIGURE 8. The simulation results of multi-objective optimization function analysis experiment. As can be seen from the path planning simulation results in Figure 8, compared with the original algorithm, the MSPGPP algorithm's solution to the multi-objective optimization function in the four scenarios results in smoother optimal paths, overcoming the drawback of excessive deflections in the original algorithm's path. The algorithm also meets the UAV's maneuverability constraints, allowing the UAV to maintain a safe distance from obstacles, effectively avoid them, and reach the target point.

three-dimensional map model with a boundary of *MAP* in three dimensions. Related experimental parameters are shown in Table 1.

The comparison experiment process is shown in Figure 5, where red obstacles represent the threat sphere model.

Comparison experiment results are shown in Figure 6.

Overall, the improved algorithm satisfies the UAV's maneuverability constraints better than the original algorithm. The results of these model and strategy analysis experiments validate the effectiveness and applicability of the improved algorithm's model and strategies.

2) MULTI-OBJECTIVE OPTIMIZATION FUNCTION ANALYSIS

This section describes the steps of the MSPGPP algorithm in solving the multi-objective optimization function and compares the MSPGPP algorithm with the PGPP algorithm. Based on the design and analysis of the MSPGPP algorithm, the steps of solving the MSPGPP algorithm under the constraint of the multi-objective optimization function are shown in Figure 7.

Parameters for the multi-objective optimization function analysis experiment are shown in Table 2, with the algorithm iteration limit set to *MaxIt*. Except for the obstacle settings, other environmental parameters are consistent with those in Table 1.

To validate the effectiveness and applicability of the improved algorithm's multi-objective optimization function, four benchmark scenarios are set, with obstacle parameter information in the scenarios shown in Table 3.

Path planning results are shown in Figure 8.

Path planning cost optimization results are shown in Figure 9.

B. COMPARISON WITH OPTIMIZATION ALGORITHMS

To further validate the performance of the MSPGPP algorithm proposed in this paper, this section compares MSPGPP

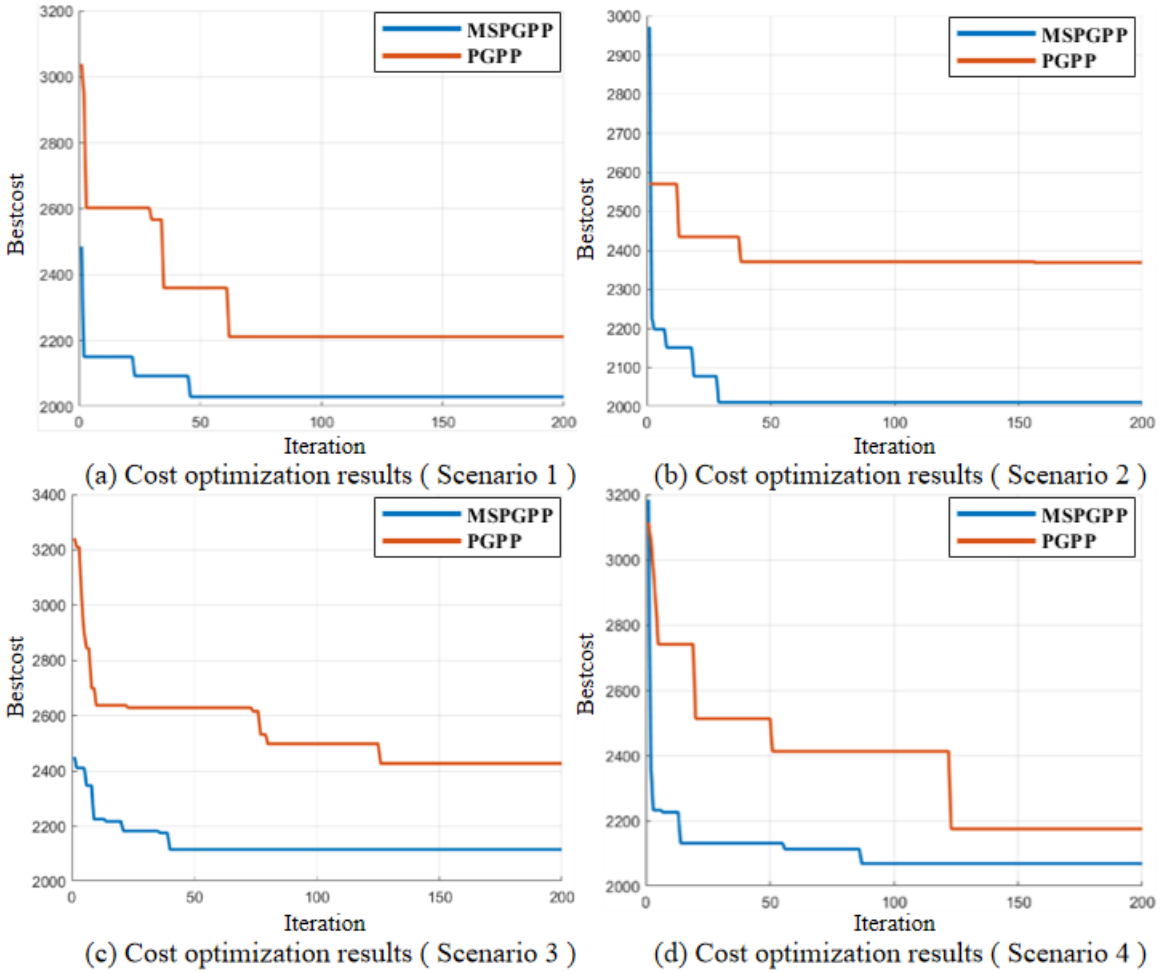

FIGURE 9. Path cost optimization results of multi-objective optimization function analysis experiment. As can be seen from the path cost optimization results in Figure 9, the MSPGPP algorithm converges more quickly in the four scenarios, while the original algorithm, due to greater randomness, converges more slowly and unstably, verifying the effectiveness and efficiency of the algorithm proposed in this paper.

TABLE 2. Experimental parameters of multi-objective optimization function analysis.

Experimental parameters	Set value
$MaxIt$	200
φmax	45°
θmax	45°
h_k	0.5
h_{max}	400m
h_{min}	100m
K_t	0.6
K_c	0.4

TABLE 3. The obstacle parameters of the original algorithm comparison experiment.

Scenario	Obstacle Center	Radius
1	(400,450,300)	35m
2	(400,450,300)	25m
	(480,370,300)	25m
	(560,290,300)	25m
3	(400,450,300)	25m
	(480,370,300)	25m
	(560,290,300)	25m
	(300,300,300)	30m
4	(550,440,300)	35m
	(450,550,300)	25m
	(360,400,300)	25m
	(650,600,300)	20m

with other advanced optimization algorithms, including the RRT algorithm, Particle Swarm Optimization (PSO) algorithm, and Grey Wolf Optimizer (GWO). The iteration limit for all four algorithms is 50. The RRT algorithm has a step size of 25 and a pathfinding limit of 5000; the PSO algorithm has a particle swarm size of 500, an iteration limit of inertia weight of 1, inertia mass damping ratio of 0.98, and individual and social learning factors of 1.5; the GWO algorithm has a wolf pack size of 20 and a maximum search step size of 30.

1) SCENARIO SETTING

Simulation experiment scenarios use the map model and experimental parameters from section A. Obstacle parameters in the environment are shown in Table 4.

TABLE 4. The obstacle parameters of optimization algorithm contrast experiment.

Num	Obstacle Center	Radius
1	(550,380,280)	35m
2	(650,500,350)	35m
3	(440,350,350)	35m
4	(450,480,350)	35m
5	(550,550,350)	35m
6	(350,450,350)	35m
7	(300,300,300)	35m
8	(770,650,300)	35m

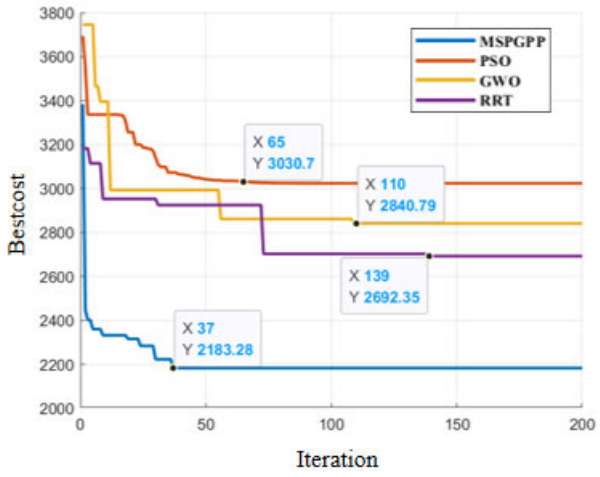

FIGURE 10. Path cost optimization results of multi-objective optimization function analysis experiment. As shown in Figure 10, the total optimal path cost planned by the MSPGPP algorithm is only 2183.28, while the GWO algorithm, PSO algorithm, and RRT algorithm all have total optimal path costs exceeding that of the MSPGPP algorithm. The MSPGPP algorithm achieves the required optimal path with only 37 iterations, while the GWO algorithm, PSO algorithm, and RRT algorithm require 110, 65, and 139 iterations, respectively, to converge. Therefore, the MSPGPP algorithm has the best effect in solving the optimal path.

2) COMPARISON OF OPTIMIZATION ALGORITHM EFFECTS

Path cost comparison optimization results are shown in Figure 10.

The algorithm optimization effect comparison is shown in Table 5.

TABLE 5. The obstacle parameters of optimization algorithm contrast experiment.

Algorithm	Computation time	Iterations	Optimal path cost
MSPGPP	34.2s	37	2183.28
PSO	149.04s	65	3030.70
GWO	747.96s	110	2840.79
RRT	26.56s	139	2692.35

As further shown in Table 5, the total duration of the improved algorithm's solution process is 34.2 seconds, while the GWO algorithm, PSO algorithm, and RRT algorithm planning total durations are 747.96 seconds, 149.04 seconds, and 26.56 seconds, respectively. The improved algorithm's timeliness is better than that of the PSO and GWO algorithms. Since the RRT algorithm has the lowest computational complexity, its solution time is the shortest, but the MSPGPP has better convergence than the RRT algorithm and plans the lowest total cost-optimal path. Combining the number of iterations of the optimal path and the total path cost, it can be seen that the MSPGPP algorithm has the best effect in solving the multi-objective optimization function, planning the most optimal path.

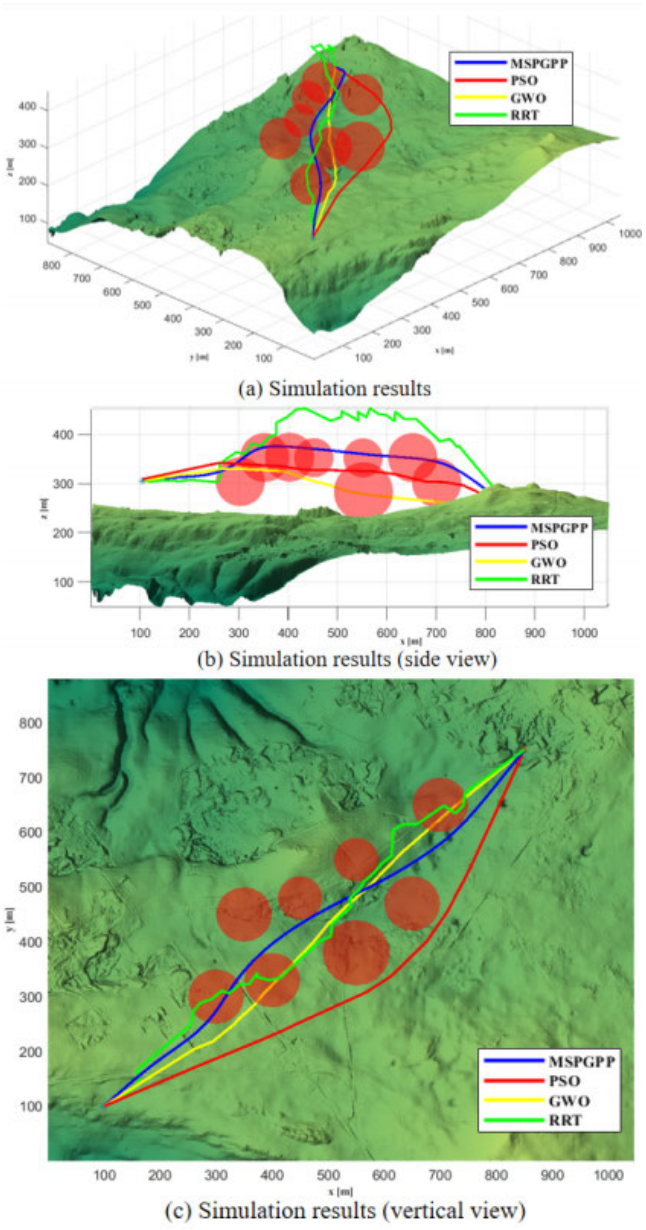

FIGURE 11. The simulation results of the optimization algorithm comparison experiment.

Optimization algorithm comparison experiment simulation results are shown in Figure 11. From the path planning simulation results, it can be seen that overall, MSPGPP, PSO, and GWO algorithms can generate feasible paths that meet constraints such as path length, flight altitude, threat level, and maneuverability, but the RRT algorithm is insufficient in considering UAV maneuverability constraints.

From the top view of the simulation results, it can be further seen that the MSPGPP algorithm can obtain the most optimal solution that takes into account path length, flight altitude, obstacle avoidance safety, and UAV maneuverability constraints. The path planned by the PSO algorithm avoids obstacles with the largest span, leaving the mission path for the longest time, resulting in high path length costs. The path planned by the RRT algorithm has excessive deflections and dispersed nodes, resulting in high maneuverability costs. The path planned by the GWO algorithm is difficult to maintain a safe distance to avoid obstacles, resulting in high threat level costs.

Figures that are meant to appear in color, or shades of black/gray. Such figures may include photographs, illustrations, multicolor graphs, and flowcharts. For multicolor graphs, please avoid any gray backgrounds or shading, as well as screenshots, instead export the graph from the program used to collect the data.

V. CONCLUSION

In response to the problem of three-dimensional path planning for Unmanned Aerial Vehicles (UAVs), this paper proposes a new MSPGPP algorithm. The simulation results of the algorithm demonstrate:

The MSPGPP algorithm defines and introduces a threat sphere model to ensure the safety of UAV obstacle avoidance. The obstacle avoidance cone model is also defined and introduced, improving the excessive randomness of the original algorithm's random branching step and enhancing computational efficiency, meeting the timeliness requirements of UAV path planning algorithms. The introduction of B-spline curves enhances path smoothness, meeting the maneuverability requirements of UAVs, overcoming the drawback of excessive deflection in the original algorithm's paths, and ensuring UAV flight safety.

The MSPGPP algorithm defines and introduces a multi-objective optimization function, taking into account various target constraints such as flight altitude and threat level, reducing path costs, and improving the deficiency of the original algorithm which did not consider path optimality.

Through three-dimensional simulation experiments, comparing the MSPGPP algorithm with the original PGPP algorithm and other optimization algorithms, the results show: Compared with the original PGPP algorithm, the MSPGPP algorithm can better balance obstacle avoidance safety and UAV maneuverability, planning paths that more safely and effectively avoid obstacles. The MSPGPP algorithm has higher accuracy and faster convergence. Compared with other optimization algorithms, including the GWO algorithm, PSO algorithm, and RRT algorithm, the MSPGPP algorithm is the most effective in solving the optimal path.

Future work of this paper will focus on applying the MSPGPP algorithm to multi-UAV formation path planning problems. The applicability of the MSPGPP algorithm will continue to be explored, to evaluate its effectiveness in different domains of optimization problems.

REFERENCES

[1] B. Li, Z.-P. Yang, D.-Q. Chen, S.-Y. Liang, and H. Ma, "Maneuvering target tracking of UAV based on MN-DDPG and transfer learning," *Defence Technol.*, vol. 17, no. 2, pp. 457–466, Apr. 2021.

[2] Y. Wang, P. Bai, X. Liang, W. Wang, J. Zhang, and Q. Fu, "Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms," *IEEE Access*, vol. 7, pp. 105086–105099, 2019.

[3] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowl.-Based Syst.*, vol. 158, pp. 54–64, Oct. 2018.

[4] X. Yang, R. Wang, and T. Zhang, "Overview of intelligent optimization algorithms for UAV swarm path planning," *Control Theory Appl.*, vol. 37, pp. 2291–2302, Nov. 2020.

[5] R. Zhang, Y. Wu, L. Zhang, C. Xu, and F. Gao, "Autonomous and adaptive navigation for terrestrial-aerial bimodal vehicles," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3008–3015, Apr. 2022.

[6] D. Chao and W. U. Qihui, "Survey of layered architecture in large-scale FANETs," *Comput. Sci.*, vol. 47, pp. 226–231, Sep. 2020.

[7] C. Zhao, Y. G. Liu, and L. Chen, "Status and prospects of multi-UAV path planning for meta-heuristic algorithms," *Control Decis.*, vol. 37, pp. 1102–1115, May 2022.

[8] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, pp. 270–299, Jan. 2020.

[9] Y. He, T. C. Hou, and Z. W. Zeng, "UAV path planning based on improved A* and dynamic window method," (in Chinese), *Mech. Sci. Technol. Aerosp. Eng.*, 2023. Accessed: Oct. 19, 2023, doi: 10.13433/j.cnki.1003-8728.20230322.

[10] C. K. Jiang, Z. Li, S. B. Pan, and Y. Wang, "Collision-free path planning of AGVs based on improved Dijkstra algorithm," *Comput. Sci.*, vol. 47, pp. 272–277, Aug. 2020.

[11] Y. Min, L. Jianjun, W. Mingming, and G. Dengwei, "Coordinated path planning by integrating improved RRT and quartic spline," *Chin. J. Mech.*, vol. 52, pp. 1024–1034, Apr. 2020.

[12] W.-H. Liu, X. Zheng, and Z.-H. Deng, "Dynamic collision avoidance for cooperative fixed-wing UAV swarm based on normalized artificial potential field optimization," *J. Central South Univ.*, vol. 28, no. 10, pp. 3159–3172, Oct. 2021.

[13] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107376.

[14] Y. Cao, W. Wei, Y. Bai, and H. Qiao, "Multi-base multi-UAV cooperative reconnaissance path planning with genetic algorithm," *Cluster Comput.*, vol. 22, no. S3, pp. 5175–5184, May 2019.

[15] X. Q. Zhang, L. Huang, and Y. T. Shi, "Improved ant colony path planning algorithm based on multi-objective optimization," (in Chinese), *Mod. Manuf. Eng.*, vol. 11, pp. 40–46, Nov. 2023, doi: 10.16731/j.cnki.1671-3133.2023.11.006.

[16] D. Sui, Z. Y. Yang, and S. B. Ding, "Three-dimensional path planning of UAV based on EMSDBO algorithm," (in Chinese), *Syst. Eng. Electron.*, 2023. Accessed: Dec. 5, 2023. [Online]. Available: http://kns.cnki.net/kcms/detail/11.2422.TN.20231205.1202.016.html

[17] R. Jarray, M. Al-Dhaifallah, H. Rezk, and S. Bouallègue, "Parallel cooperative coevolutionary grey wolf optimizer for path planning problem of unmanned aerial vehicles," *Sensors*, vol. 22, no. 5, p. 1826, Feb. 2022.

[18] Z. Q. Liu, L. He, and L. Yuan, "Mobile robot path planning based on improved Grey Wolf algorithm," *J. Xi'an Jiaotong Univ.*, vol. 56, pp. 49–60, Oct. 2022.

[19] X. L. Chen, Y. J. Huang, and Q. Fan, "Three-dimensional path planning of UAV based on multi-modal multi-objective evolutionary algorithm," *Fire Control Command Control*, vol. 48, pp. 32–39, Nov. 2023.

[20] N. Lin, J. Tang, X. Li, and L. Zhao, "A novel improved bat algorithm in UAV path planning," *Comput., Mater. Continua*, vol. 61, no. 1, pp. 323–344, 2019.

[21] J. Huang, Y. F. Li, and S. C. Wang, "Three-dimensional path planning of UAV metropolitan area based on improved PSO algorithm," *Electron. Opt. Control*, vol. 31, no. 2, pp. 41–45, 2024. Accessed: Jul. 4, 2023.

[22] Z. F. Huang and Y. H. Liu, "Research on UAV path planning based on improved Harris Hawk and B-spline curve," (in Chinese), *J. Syst. Simul.*, 2023. Accessed: Jun. 8, 2023, doi: 10.16182/j.issn1004731x.joss.23-0403.

[23] Y. Zhou, Y. Su, A. Xie, and L. Kong, "A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV," *Chin. J. Aeronaut.*, vol. 34, no. 9, pp. 199–209, Sep. 2021.

[24] Y. Zhou, Y. Wang, X. Chen, L. Zhang, and K. Wu, "A novel path planning algorithm based on plant growth mechanism," *Soft Comput.*, vol. 21, no. 2, pp. 435–445, Jan. 2017.
[25] F. F. Xi, X. Zeng, and S. M. Ji, "Mobile robot path planning based on PG-RRT algorithm," *Comput. Sci.*, vol. 46, pp. 247–253, Apr. 2019.
[26] H. Guo and X. H. Guo, "Local path planning algorithm for UAV based on improved velocity obstacle method," *Acta Aeronautica Astronautica Sinica*, vol. 44, pp. 271–281, Nov. 2023.
[27] L. Sun, L. S. Zhang, and L. Zhou, "A new method for mobile robot trajectory planning based on Bezier curve," *J. Syst. Simul.*, vol. 30, pp. 962–968, Mar. 2018.

HENG XIAO received the Ph.D. degree in management science and engineering from Harbin Engineering University, in 2015. From 2015 to 2021, she was a Lecturer with the Faculty of Civil Aviation and Aeronautics, Kunming University of Science and Technology, Kunming, China. Since 2021, she has been an Associate Professor with the Faculty of Civil Aviation and Aeronautics, Kunming University of Science and Technology. She is the author of two books, more than ten articles, and four inventions. Her research interests include path optimization, operation control management, and innovation management.

ZHENJIE MU was born in Xi'an, Shaanxi, China, in 1998. He received the B.S. degree in management from Xi'an University of Posts and Telecommunications (XUPT), in 2021. He is currently pursuing the M.S. degree in transportation with Kunming University of Science and Technology (KUST). His research interests include UAV path planning, UAV obstacle avoidance, and multi-UAV cooperative technologies.

WEN ZHOU received the Ph.D. degree in management science and engineering from Harbin Engineering University, in 2015. He is currently an Associate Professor with the College of Systems Engineering, National University of Defense Technology, Changsha, China. His current research interests include unmanned system machine intelligence and complex networks.

HUI ZHANG was born in Kunming, Yunnan, China, in 1995. She received the B.S. degree in automotive service engineering from Kunming University of Science and Technology (KUST), in 2017, where she is currently pursuing the M.S. degree in transportation. Her research interests include UAV path planning and formation control technologies.