Journal of Systems Engineering and Electronics
Vol. 37, No. 3, June 2026, pp.1042 – 1058

Three-dimensional path planning algorithm for UAV based on obstacle envelopes

ZHOU Yuzhen[1], LIU Yao[2], HUANG Jincai[2], and SHI Jianmai[2,*]

1. School of Data and Target Engineering, Information Engineering University, Zhengzhou 450000, China;
2. College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

Abstract: In this paper, a three-dimension envelope-based path planning algorithm (3DE-PP) is proposed to automatically generate a collision-free trajectory for unmanned aerial vehicle (UAV). Firstly, focusing on the defects of low efficiency of obstacle modelling representation and large search space, an elliptical envelope-based obstacle modelling method is proposed to facilitate the generation of obstacle avoidance waypoints and improve the search efficiency. Then, considering safety and aiming at minimum energy consumption, waypoint generation strategies based on tangent guidance and minimum deviation are designed. Meanwhile, aiming at the UAV motion constraint, a three-dimension (3D) path construction method based on improved Dubins is proposed. Finally, combined with the main path generation algorithm based on saving algorithm, a safe and feasible 3D flight path is constructed by considering the power constraint of UAV and the access of charging stations comprehensively. The proposed 3DE-PP is compared with four algorithms (SAS, Dubins-RRT*, APF, 3D-TG) by 15 examples generated from five typical environments, and the computational results confirm its advantages. Furthermore, a real-world case is introduced, and the key factors influencing path planning are analyzed.

Keywords: ellipsoidal envelope, unmanned aerial vehicle (UAV), path planning, obstacle avoidance.

DOI: 10.23919/JSEE.2026.000126

1. Introduction

Due to the great advantages in motility, cost and casualties [1,2], small unmanned aerial vehicles (UAVs), also named as drones, play an important role in a variety of fields, such as cargo delivery [3], surveillance and security [4], agriculture [5], communications [6], infrastructure inspections [7], along as well as search and rescue [8,9]. Many companies, such as Amazon [10], Google Wing [11], JD.com [12] and DHL [13], have also widely explored the applications of UAVs, especially in logistics. The UAVs were first introduced in e-commerce delivery in 2013 [14]. In 2019, united parcel service (UPS) [15] applied drones for the delivery of medical samples in North Carolina. Lee [16] proposed the applications of drones in food and beverage deliveries to avoid congested urban traffics. The convenience of UAVs has become increasingly evident, particularly since the advent of the corona vius disease 2019 pandemic in 2020. UAVs can transport supplies without direct contact, which significantly mitigates the risk of viral transmission and concurrently boosts operational efficiency. Path planning in uban areas becomes a critical issue for UAV delivery.

In 3D urban space, UAVs need to traverse dense obstacles and narrow corridor spaces, and the difficulty of path planning increases exponentially due to range limitations and increasingly complex kinematic constraints, a lot of research has been carried out in order to plan a collision-free path in complex environments.

Path planning methodologies currently in use can be categorised into five distinct types [17]: graph-based methods, sampling-based methods, potential field methods, intelligent optimisation methods, and artificial intelligence-based methods.

In dense urban obstacle environments, the majority of graph-based and intelligent optimization techniques rely on traversing points to find paths, thereby increasing time costs. Alternatively, methods based on tangents identify paths by establishing tangents between obstacles. This approach can reduce time costs. However, current research in this field primarily remains confined to 2D space.

The primary challenges of path planning within 3D space encompass several key considerations. Firstly, the modeling of architectural obstacles in low-altitude environments is crucial to support effective path planning. Secondly, the higher degree of freedom within the 3D space results in a greater number of collision-free paths,

Manuscript received November 18, 2024.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (72271241).

which complicates the search process. Consequently, it is necessary to design logical path generation and search strategies. Finally, the manieuvrability of UAV, including factors such as minimum turning radius and pitch angle, must be taken into account within the 3D space.

This paper investigates the path planning problem in 3D space. Current obstacle modelling methods suffer from low representation efficiency and a vast search space. To counter these issues, we propose an obstacle modelling technique based on elliptical envelopes. Subsequently, three collision-free paths are generated from both sides and above the obstacle, reducing large-scale searches of the entire space. Furthermore, a path smoothing method based on the modified Dubins' model is proposed to consider the UAV motion constraints, e.g., minimum turning radius and maximum climb rate. Meanwhile, the main path generation algorithm using a saving method is presented with consideration of the UAV range and charging stations availability.

The key contributions of this study are summarized as follows:

(i) A 3D envelope-based environmental modeling method is proposed to model obstacles as combinations of elliptic columns and ellipsoids, simplifying the creation of obstacle avoidance waypoints.

(ii) A tangent-guided approach, coupled with a minimum deviation strategy, is introduced to generate three collision-free paths around obstacles when they are encountered, which ensure the fast generation of collision-free paths, select one of the paths according to the designed heuristic rules, and repeat the above operation until the destination is reached.

(iii) A method for calculating angles is developed according to UAV flight dynamics, in the context of a 3D Dubins algorithm, aiming to reduce hovering during take-off and landing of UAVs.

(iv) The efficacy of the 3D envelope-based path planning (3DE-PP) algorithm has been corroborated through 15 randomly generated cases in five classical environments, and a real-world scenario. The experimental results indicate that 3DE-PP is capable of efficiently generating high-quality, collision-free paths in a variety of obstacle-dense environments.

This paper is structured as follows. Section 2 reviews relevant research. The problem and the modeling method for obstacles in 3D environment is introduced in Section 3. Then, Section 4 presents the 3DE-PP algorithm. To test the performance of algorithm, contrast experiment on stochastic examples is conducted in Section 5, as well as sensitivity analysis on real instances. Finally, Section 6 gives conclusions and outlines future work.

2. Literature review

Path planning for UAVs refers to forming a feasible and collision-free path from the starting point to the endpoint in a complex environment with obstacles, based on evaluation criteria such as path length, energy consumption, and smoothness [18]. Detailed discussions on 3D UAV path planning and obstacle avoidance technology could be found in [19–21], while Phalapanyakoon et al. [22] provided comprehensive insights into the modeling and representation of the environment. This section presents a brief overview of the current research status of UAVs.

UAV path planning methods are classified as graph-based, sampling-based [23], potential field methods [24], intelligent optimization methods [25] and artificial intelligence methods [26].

Szczerba et al. [27] introduced a graph-based technique, sparse A* search (SAS), designed to efficiently identify collision-free paths in 3D environments. Subsequently, Liu et al. [28] and Zhang et al. [29] formulated a 3D jump-point search, which enhances the search speed by pruning the neighbors of the nodes under consideration. This approach significantly reduces computational overhead compared to the traditional A* method.

Rohnert [30] first proposed a tangent method to find paths based on the common tangent of convex polygons. Liu et al. [31] studied path planning using a tangent graph between polygonal and curved obstacles and proved that the tangents are locally shortest straight-line paths. Liu et al. [32] constructed the entire tangent map before path planning. In order to overcome the time-consuming and large storage requirements for calculating the common tangent of polygonal obstacles, Chen et al. [33] modelled obstacles as circles. To better characterize the narrow barriers and corridors, Liu et al. [34] modelled the obstacles as ellipses. Although ellipses help with route smoothing, the entire roadmap still needs to be built in a tangent map, which hinders real-time planning. Thus, Liu et al. [35] proposed an autonomous path planning algorithm based on tangent intersection and target guidance strategies for 2D environments. Yang et al. [36] proposed a 3D path planning method based on convex polyhedra. Liang et al. [37] and Liu et al. [32] investigated a cube-based 3D UAV path planning.

Rapid exploration random tree (RRT) [38] is the most commonly used sampling-based method, but it has been proven not to be optimal [39]. Therefore, Gammell et al. [40] proposed Informed RRT* to overcome this and speed up the search by introducing tree increments and limiting the search range.

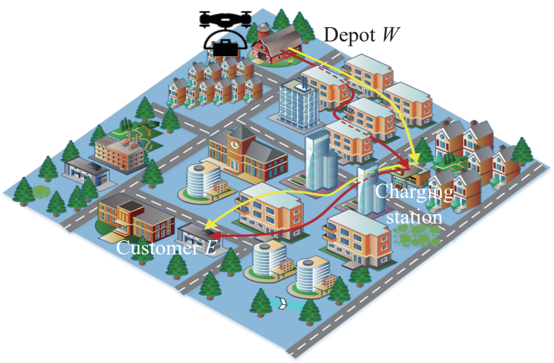
Artificial potential fields (APFs) [41] perform well in path planning in general environments, while they may fall into local optima in dense cluttered environments.

Intelligent optimisation methods have received a lot of attention due to their global optimisation capabilities. Roberge et al. [42] used genetic algorithm (GA) and particle swarm optimisation algorithms to generate feasible paths in 3D environments. Wu et al. [43] investigated 3D path planning based on ant colony algorithm. In recent years, deep learning [26] and reinforcement learning [44] have also been applied to 2D path planning. Motions such as height adjustments and turns increase energy loss, so smoothing the path as much as possible can reduce energy consumption. Among these methods, the B-splines [45] and dubins methods [46,47] are usually used to smooth the paths.

From the above review of the related literature, we can find that: (i) To the best of our knowledge, no researcher has so far successfully extended the elliptic tangent-based approach from 2D to 3D. (ii) It is still a challenge to balance time and solution quality in 3D path planning.

3. Problem formulation

In the example presented in Fig. 1, the UAV is used to deliver cargo from depot W to customer E. The multiple crowed buildings in the environment and the limitation of UAV endurance prevent the UAV from flying directly from W to E. Therefore, it is necessary to plan a collision-free and flyable path from point W to point E. The red curve in the diagram is the traditional 2D obstacle avoidance path, which is planned by assuming that the UAV is flying at a fixed altitude. The UAV has to navigate between the crowded buildings and achieve obstacle avoidance by changing its flight direction several times. The yellow curve shows a 3D obstacle avoidance path where the UAV can adjust its flight altitude according to the height of the obstacle. For some dense obstacles that do not exceed the maximum flight altitude of the UAV (as shown in the top right corner of Fig. 1), the UAV can fly over the building after raising its flight height, which obviously consumes less energy.



: Fly-over path; : Fly-around path.

Fig. 1 UAV delivery and charging scenarios

Besides, since the energy consumption from the depot W to the customer E exceeds the capacity of UAV battery, it is supposed to select an appropriate charging station, where the UAV can be recharged for successful delivery.

Compared to 2D planar obstacle avoidance path planning, 3D path planning needs to consider the motion constraints of the UAV in both horizontal and vertical directions, i.e., the projection of the UAV's turning radius in 3D space must not exceed the minimum turning radius in the horizontal direction and must not violate the maximum climb rate constraint in the vertical direction. This integrated horizontal and vertical control strategy in 3D space makes it more difficult to solve for the UAV's flyable path. In addition, the UAV has to consider the maximum flight altitude and maximum range in flight. In order to describe these types of UAV constraints more clearly, the modelling of UAV motion constraints is presented in the following subsection.

3.1 Motion constraints of UAV

To obtain a feasible and flyable UAV path, the following UAV motion constraints must be satisfied. The notations used to character the UAV are listed in Table 1.

Table 1 Definition of notations describing UAV

Notation	Description
$Y_{\max}$	Maximum UAV power
P_i	The waypoint
$E(P_i,P_j)$	The energy consumption between P_i and P_j
m_t	The net weight of the drone
m_d	The weight of the battery
m_1	The weight of the cargo
θ_v	The lift-to-drag ratio, related to speed
η	The energy transfer efficiency
g	The gravitational acceleration
v	The flight speed
$R_{\min}$	The minimum turning radius
$R_{\max}$	The limb rate
r	The minimum safety distance
$H_{\max}$	The maximum altitude
L_i	The obstacles $(i=1,2,\cdots,N)$
x_i,y_i,z_i	The coordinates of the centre point of the obstacle
l_i,w_i,h_i	The length, width and height of the obstacle

(i) Maximum endurance $Y_{\max}$: The maximum endurance capacity $Y_{\max}$ must be taken into account during the planning of the UAV path due to its limited battery power. If the energy consumption to the next customer point exceeds the maximum endurance, the UAV needs to visit an appropriate charging station to get recharged. Assuming that there are M points $(P_1, P_2, \cdots, P_M)$ in the flight path, then the energy consumption y_i when the UAV reaches point P_i $(1 \leqslant i \leqslant M)$ can be calculated as

$$\begin{cases} y_i = y_{i-1} + E(P_{i-1}, P_i), & P_{i-1} \text{ is a customer point} \\ y_i = E(P_{i-1}, P_i), & P_{i-1} \text{ is a rechaging station} \end{cases}, \quad (1)$$

where the flown energy capacity would be reset when the UAV visits a charging station.

Thus, the UAV fight capacity y_i must satisfies the following constraints: $y_i \leqslant Y_{\max}$, where $E(P_i, P_j)$ represents the energy required between P_i and P_j and can be calculated by following [48]:

$$E(P_i, P_j) = \int_{P_i}^{P_j} e_{\frac{c}{d}} \mathrm{d}s \quad (2)$$

where s denotes the distance and $e_{\frac{c}{d}}$ is the energy consumption per unit length which is related to the UAV characteristics and the position of P_i and P_j. Specifically, the subscript c denotes the cruising state of the UAV, while d refers to the flight distance. When P_i and P_j are at the same height,

$$e_{\frac{c}{d}} = \frac{(m_t + m_d + m_1)g}{3\,600\theta_v \eta},$$

otherwise:

$$e_{\frac{c}{d}} = \frac{(m_t + m_d + m_1)g}{3\,600\theta_v \eta} + \frac{(m_t + m_d + m_1)gv}{\eta}.$$

Therefore when the UAV is flying at a constant speed, its energy consumption during climb differs from level flight by a constant.

(ii) Minimum turning radius $R_{\min}$ & climb rate $R_{\max}$: Both rotary-wing and fixed-wing UAV cannot instantly change their flight direction, and can only achieve turning under the limitation of the minimum turning radius $R_{\min}$. $R_{\min}$ is the radius of the UAV in the horizontal plane during its extreme overload circular motion, which is calculated as $R_{\min} = v^2/(g \tan \phi_{\max})$, where $\phi_{\max}$ indicates the maximum roll angle of the drone. Climb rate is the height that the UAV rises in unit time, and the climb rate is mainly affected by the climb angle, so this paper limits the UAV's climb rate by limiting the minimum turning radius of the UAV in the vertical horizontal plane. Therefore, these two constraints must be considered while planning the 3D flight path of UAV.

(iii) Maximum altitude $H_{\max}$ and minimum safety distance r: Due to urban policies and aviation system regulations, UAVs are restricted in their flight altitude and have a maximum flight altitude. For example, in China, light and micro-UAVs can only fly below 200 m and 50 m, respectively. In addition, drone flight areas should be spaced apart from public buildings.

The mathematical description of the problem is as follows. Given the planning space A, the starting point W and the end point E are pre-known, and there exist some obstacles $L_1, L_2, \cdots, L_N$ and optional charging stations $\mathrm{CS}_1, \mathrm{CS}_2, \cdots, \mathrm{CS}_M$ in space A. The minimum safe distance between the UAV and the obstacle is denoted as r. The problem is to find a feasible collision-free path for the UAV, in which one or multiple charging stations could be visited for recharging, so as to minimize the overall energy consumption without violating the endurance and kinematic constraints.

In terms of modeling obstacles, as the non-uniform shape of obstacles would lead to expensive computations and unsmooth paths, many methods would model obstacles as regular geometry with smooth surfaces. However, different from the existing methods treating obstacles as ellipses [34], this paper would model the obstacles as a combination of ellipsoid above and elliptical cylinder below, which would better facilitate the construction of the UAV path to avoid obstacles in different directions. Formally, the center point of the obstacle L_i is recorded as (x_i, y_i, z_i), and the length, width and height of the obstacle are denoted as $2l_i$, $2w_i$ and h_i, respectively. Detailed modeling method of obstacles is presented in Subsection 3.2.

3.2 Modeling obstacles in 3D environment

To optimize the computational cost for the UAV path, all buildings are modeled as uniform regular geometry in Fig. 2, in which the upper part is a half ellipsoid and the lower is elliptic cylinder. Other obstacles such as trees and poles can be modelled similarly.

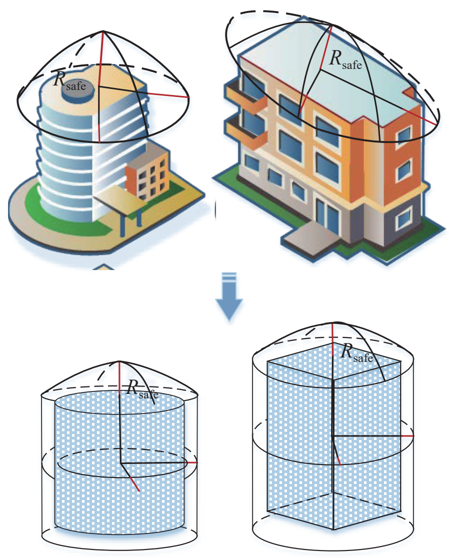


Fig. 2 Envelope of obstacles

At first, the envelope is built for each obstacle, where the longest width of the obstacle is calculated as the long axis of the elliptic cylinder and the obstacle length perpendicular to this direction is the short axis of the elliptic cylinder correspondingly.

Then, to further simplify the calculation, when the flight height z is less than the height h_i of the obstacle L_i, which means there is no conflict, the obstacle can be directly considered as an ellipse, and the elliptic equation is

$$\frac{((x-x_i)\cos\theta+(y-y_i)\sin\theta)^2}{(a+r)^2}+\frac{((y-y_i)\cos\theta-(x-x_i)\sin\theta)^2}{(b+r)^2}=1,\ z<h_i \tag{3}$$

where θ is the inclination of the semi-major axis.

Conversely, if the UAV can successfully fly over the obstacle, i.e., $z \geqslant h_i$, the formula of the obstacle is

$$\frac{((x-x_i)\cos\theta+(y-y_i)\sin\theta)^2}{(a+r)^2}+\frac{((x-x_i)\cos\theta+(y-y_i)\sin\theta)^2}{(a+r)^2}+\frac{(z-h_i)^2}{r^2}=1,\ z\geqslant h_i. \tag{4}$$

For the above two cases, each point $P_j(x_j,y_j,z_j)$ on a collision-free path must satisfy the following constraint:

$$\frac{((x-x_i)\cos\theta+(y-y_i)\sin\theta)^2}{(a+r)^2}+\frac{((y_j-y_i)\cos\theta-(x_j-x_i)\sin\theta)^2}{(b+r)^2}\geqslant 1,\ z<h_i; i=1,2,\cdots,N, \tag{5}$$

$$\frac{((x_j-x_i)\cos\theta+(y_j-y_i)\sin\theta)^2}{(a+r)^2}+\frac{((y_j-y_i)\cos\theta-(x_j-x_i)\sin\theta)^2}{(b+r)^2}+\frac{(z_j-h_i)^2}{r^2}\geqslant 1,\ z\geqslant h_i,\ i=1,2,\cdots,N. \tag{6}$$

4. Algorithm design

Traditional graph-based path planning algorithms generally consist of two main steps [49]: step one is to create a road map and step two is to search for high-quality paths on this map. To search for high-quality paths, they usually generate many candidate collision-free paths from the starting point to the target on the roadmap.

With large storage space and low search efficiency, this kind of methods is not applicable when the environment expands from 2D plane to 3D space. Thus, a 3DE-PP method is proposed. When an obstacle is encountered, instead of constructing an entire roadmap, the method generates three sub-path points based on the constructed obstacle envelope by considering both fly-around and fly-over via a tangent guidance strategy and a minimum deviation strategy, respectively. Sub-paths are generated based on the improved 3D Dubins path construction method, and then design rules are used to select one of the sub-paths considering the obstacle avoidance conditions and sub-path energy consumption. The UAV flies along the sub-path and iteratively adjusts its flight path to avoid obstacles until the collision-free path extends to the target. Based on this, combining the savings algorithm-based main path construction method until the collision-free path is extended to the target forms the final collision-free path.

4.1 Notations

To better describe the 3DE-PP algorithm, the notations involved in the algorithm is listed in Table 2. Note that the set of points is S, containing customer points and charging stations, and the set of obstacles is L. Referring to the definition in [7], we define the tangents from the starting point or the end point to the obstacle as the starting point tangent or the end point tangent, respectively. These tangent lines would generate different waypoints. Through judging whether paths through different waypoints are feasible, two sets are constructed: the points in the set Pa are flyable points, which can be connected through the Dubins curves to determine the final collision-free flyable path, while the set Ca records the infeasible points which require further planning. For example, the tangent to the ellipse through the starting point O and the ending point D can generate two fly-around points, and the straight line through OD could determine a fly over point, one of which would be selected as a waypoint T based on heuristic rules. If there is no collision in the Dubins path from point O to T, T will be added to Pa; otherwise, it will be included in the set Ca.

Table 2 Definition of notations

Notation	Description
S	Points set
L	Obstacle list
T	Generated path points
O, D	Origin and destination points, i.e., two vertices of each sub-path
Pa	Set of determined waypoints used to generate collision-free edges in order
Ca	Set of candidate waypoints
Ba	Set that records the obstacles avoided
H	The cruising altitude

4.2 3DE-PP algorithm

The 3DE-PP algorithm first generates the elliptical cylindrical envelope of the obstacle based on the model of obstacles (Line 3). Then, without the consideration of obstacles, the order of visiting customer points and charging stations is obtained with the improved saving algorithm (Line 4). Thirdly, for each adjacent two points W, E in the path (Line 6), add W to waypoints set Pa and E to the candidate waypoints set Ca (Line 7). Fourthly, while Ca is not None, repeat the following process. Determine the take-off and landing angle of the UAV, generate the Dubins paths (Lines 9–11). Check whether there are obstacles on the path (Line 13). If there is a conflict, a new waypoint is generated according to the obstacle avoidance path algorithm generateWaypoint (Line 16), and add the node T into the set of candidate points (Line 17). Finally, when a collision-free path between W, E is formed, i.e., Ca is None, add the path to Pt (Line 20). The pseudo-code of 3DE-PP can be described as Algorithm 1.

Algorithm 1 3DE-PP

1 **Input** Points set S, Obstacle list L
2 **Output** Path Pt
3 Generate the models of obstacles
4 Visit sequence Seq = Saving(S)
5 Pt = {}
6 **For** W, E in Seq:
7 Pa = $\{W\}$, Ca = $\{E\}$
8 **While** Ca is not None:
9 O = Pa(end), D = Ca(end)
10 Determine the angles of O, D, $\text{Ang}_O, \text{Ang}_D$ = Angle(O, D)
11 Build the Dubins path Path_{OD} between OD (Subsection 4.2.1)
12 Detects if there is a conflict in the path Path_{OD}
13 **If** collision-free:
14 Added D to Pa, Remove D from Ca
15 **Else**
16 T = generateWaypoint(O, D, L) (Subsection 4.2.2)
17 Added T to Ca
18 **Endif**
19 **End while**
20 Added Pa to Pt
21 **End for**

4.2.1 Improved saving algorithm

An improved saving algorithm is used to generate the initial sequence of visiting points. The saving algorithm [50], also known as the Clarke-Wright saving (CW) algorithm, is a classical constructive heuristic algorithm to solve the vehicle routing problem.

First, the CW algorithm generates paths without considering the range time constraints of UAVs. These paths are subsequently adjusted to account for the range time constraints, and their feasibility is ensured by strategically inserting charging stations using a greedy approach. we do not directly account for the linear energy consumption between waypoints when integrating charging stations. Instead, we assess the consumption based on the 3D path and introduce a specific threshold to ensure the drone's successful arrival at the charging station. For example, the 3D path energy consumption of node P_i and node P_j is its linear energy consumption $E(P_i, P_j)$ plus the energy consumption from take-off to cruise altitude and the energy consumption from cruise altitude down to ground level, to which a certain threshold is added. By this algorithm, the order of visiting customers and charging stations can be obtained.

4.2.2 Improved 3D Dubins path construction method

To generate a collision-free path that satisfies the UAV's performance constraints, an improved Dubins algorithm is used to construct 3D path, which incorporates a new flight dynamics-based angle calculation method.

The traditional Dubins path, also called the Dubins car path, is the shortest feasible path limited by the minimum curvature between two directed points in 2D plane. Unlike 2D Dubins curves which have only left and right turns in their path, 3D Dubins curves have countless turning directions, which leads to many possible paths to the final point. Several methods to obtain 3D Dubins paths have been developed, such as adding turnings [51] or an additional intermediate arc [52] in the path. Among these methods, Hota et al. [53] proposed an exact algorithm, which obtains the best flyable path between two points based on 3D spatial geometry. As shown in Fig. 3, given the starting point $O(x_O, y_O, z_O)$, the ending point $D(x_D, y_D, z_D)$ and minimum radius $R_{\min}$, the optimal path can be constructed as long as the heading angle ψ and flight path angle γ are provided.

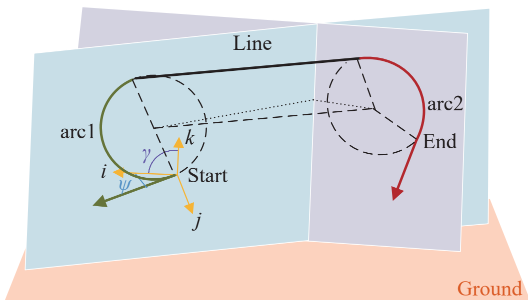


Fig. 3 Three-dimensional Dubins curves

The heading angle denotes the deviation of the path to the left and right, and is defined as the angle between the horizontal plane projection of the flight direction and the x-axis, and takes values in the range $[-\pi,\pi]$. The path flight angle, on the other hand, establishes the pitch of the aircraft. It is the angle between the path trajectory and its projection in the horizontal plane and takes values in the range $[-\pi/2,\pi/2]$. A positive angle indicates that the UAV is ascending, while a negative angle suggests that the UAV is descending. In the construction of a 3D Dubins path, if the heading angle and path flight angle do not match the direction of the next waypoint, the UAV needs to complete an aerial turn, which results in an increase in energy consumption. Furthermore, the UAV may initially and ultimately trace circles to ensure it meets the minimum flight radius and maximum climb rate, which results in varying energy consumption. Consequently, to minimize energy use during take-off and landing, this subsection proposes an angle calculation method designed to reduce unnecessary turns.

An angle algorithm based on 3D aircraft kinematic model and practical flight experience is designed to determine the flight angles, including the heading angle ψ and flight-path angle γ.

As shown in Fig. 4, the UAV needs to fly from node $O(x_O,y_O,z_O)$ to next node $D(x_D,y_D,z_D)$. To avoid unnecessary detours, in real life, aircraft generally take off directly towards the next point. Similarly, in order to avoid unnecessary turns in the generated Dubins path, the heading angle ψ and flight-path angle γ at D are determined by the direction vector $\boldsymbol{V}(\boldsymbol{x},\boldsymbol{y},z)=(x_D-x_O,y_D-y_O,z_D-z_O)$.

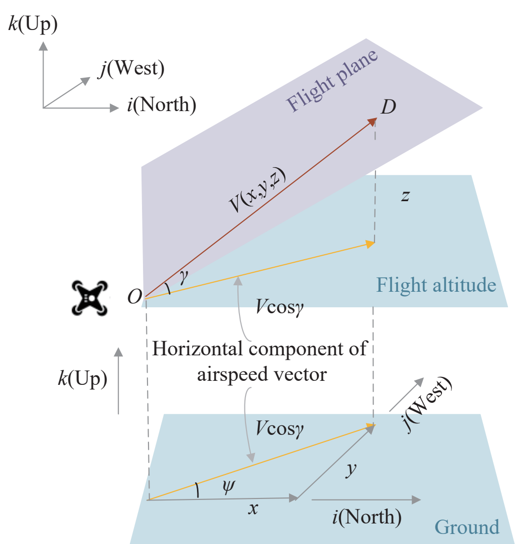


Fig. 4 Graphical representation of aircraft kinematic model

According to the dynamic equations [54], the relationship between the heading angle ψ and flight-path angle γ and the vectors is as follows:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}=\begin{pmatrix} v\cos\gamma\cos\psi \\ v\cos\gamma\sin\psi \\ v\sin\gamma \end{pmatrix} \tag{7}$$

where $v=\|\boldsymbol{V}\|$.

Thus, we can get the following formula:

$$\psi=\arctan\frac{y}{x}, \tag{8}$$

$$\gamma=\arcsin\frac{z}{v}. \tag{9}$$

According to the above equations, the heading angle ψ and flight-path angle γ at each waypoint can be determined by the predecessor node. Besides, the angle of the initial point is determined by the first point to be visited. This method of calculating angles enables the UAV to fly directly towards the target point in most instances, thereby avoiding unnecessary detours.

4.2.3 Waypoint generation

The enhanced 3D Dubins path construction technique enables the creation of a path between two unobstructed nodes. However, the path becomes unfeasible if obstacles intervene between the nodes. To circumvent these obstacles, we design a waypoint generation algorithm based on the 3D envelope of obstacles.

This algorithm is integrated with the refined 3D Dubins path construction method to produce a collision-free trajectory between the nodes.

Before introducing the waypoint generation algorithm, we delineate two methodologies for obstacle avoidance by the UAV: fly-over and fly-around. These are depicted in Fig. 5 using distinct colors, with the red line representing fly-over and the yellow line denoting fly-around. When the UAV encounters obstacles during flight, three flyable Dubins sub-paths are generated based on these avoidance strategies. The UAV then selects one of these paths according to predefined rules. This obstacle avoidance procedure is repeated until the UAV reaches its target point, culminating in a collision-free 3D Dubins trajectory from its starting point to its destination.

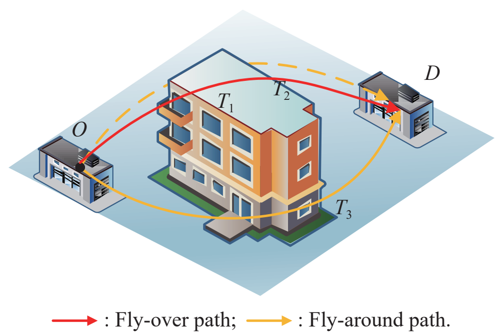


Fig. 5 Flight avoidance of obstacles

The waypoint generation algorithm is delineated in three principal sections. Initially, the fly-around points generation algorithm is devised, predicated on the principle of minimizing both the number of obstacles and energy consumption on the origin tangent and end tangent, utilizing the tangent guidance strategy. Subsequently, the fly-over points generation algorithm is formulated, rooted in the concept of adhering as closely as possible to the original path angle, employing a minimum deviation strategy. Finally, the optimal waypoint selection strategy is designed based on information such as energy consumption, number of obstacles and height. The process is repeated until a collision-free path between nodes is generated. The specific process is shown in Algorithm 2.

Algorithm 2 Generate waypoint algorithm

Input O, D, Obstacle list L
Output Waypoint set Pa
Let Pa $= O$, Ca $= D$
While Ca is not None:
  $O =$ Pa(end), $D =$ Ca(end)
  If Path$_{\mathrm{OD}}$ conflicts with an obstacle**:**
    Records of conflicting obstacles OL_1
    Calculate the optimal fly-around point $L =$ Fly_around(O, D, OL_1)
    Calculate fly-over point PQ = Fly_over(O, D, OL_1)
    Calculate optimal waypoint $T_1 =$ Waypoint selection algorithm $(L, \mathrm{PQ}, \mathrm{OL}_1)$
    Add T_1 to Ca
  Else
    Added D to Pa, Remove D from Ca
  Endif
EndWhile

(i) Generation and selection of fly-around points

Given that all obstacles have been modeled as a semi-ellipsoid in the upper section and an elliptic cylinder in the lower section, each obstacle is projected as an ellipse on the plane corresponding to the UAV's current flight height. Subsequently, two fly-around points can be generated by calculating the origin-tangent and end tangent, resulting in two distinct sub-paths.

An initial selection would be conducted between the two sub-paths. Given that both paths have an identical lift-to-drag ratio, the path length is inversely proportional to the energy consumption. Based on this conclusion, four rules would be referred:

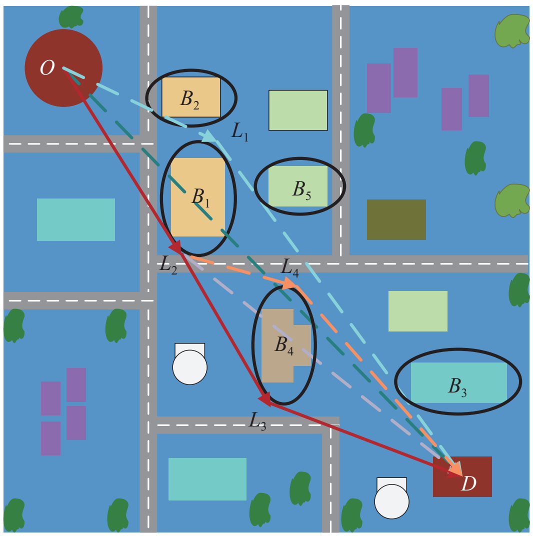
i) Rule 1: Give preference to the path with the least conflict with the origin-tangent. The existence of obstacles will certainly increase the flight distance of the UAV, so the less obstacles exist on the newly generated sub-path, the shorter the flight distance of the UAV.

ii) Rule 2: Consider the last obstacle avoided and select the path that does not conflict with it. Paths that conflict with the last obstacle need to fly in the opposite direction first to get around the obstacle, which leads to more detours and thus increases the path length.

iii) Rule 3: Choose the path with the least conflict with the end tangent. The whole path is composed of the origin-tangent and the end tangent, considering the length of subsequent paths is more conducive to obtaining the optimal path.

iv) Rule 4: The path with the shortest length is preferred. When the number of obstacles is the same, the sub-path with the shortest length should be selected to generate the final optimal path.

Fig. 6 illustrates examples of obstacle avoidance path selection. The UAV flies from node O to node D, B_1 is the first conflicting obstacle. Two fly-around paths $O \to L_1 \to D$ and $O \to L_2 \to D$ can be obtained. It is obvious that origin-tangent OL_1 would conflict with B_4 while origin-tangent OL_2 is collision-free. Thus, the sub-path $O \to L_2 \to D$ is selected according to Rule 1. However, when the UAV flies from node L_2 to node D, it conflicts with obstacle B_4 and both generated sub-paths $L_2 \to L_3 \to D$ and $L_2 \to L_4 \to D$ comply with Rule 1, in this case, the sub-path with the shortest path length $L_2 \to L_3 \to D$ is selected according to Rule 4, which results in the final path $O \to L_2 \to L_3 \to D$.



: Heuristic path 1; : Heuristic path 2; : Selected path; : Heuristic path 3; : Heuristic path 4.

Fig. 6 Selection of fly-around points

(ii) Generation of fly-over points

The determination of fly-over point coordinates is influenced by several factors: the height of the impending obstacle, the existing flight altitude, the safety distance, and the cruising altitude. By considering the height of the preceding obstacle, we can ensure that the UAV maintains a consistent altitude, thereby minimizing energy consumption fluctuations associated with varying heights.

As shown in Fig. 7, firstly, two points O', D' are obtained from the projection of O, D in the height plane of the obstacle. Then, the straight line $O'D'$ intersects with the obstacle model at P', Q'. Finally, the height coordinates of the fly over points P, Q are denoted as $\max\{h_1+r, h_p+r, H\}$. h_1 represents the height of the conflict obstacle, r denotes the safety distance, h_p stands for the current flight altitude and H is the cruising altitude. If $\max\{h_1+r, p+r, H\} < H_{\max}$, then P, Q are used as the alternative waypoints for fly-over, otherwise fly over of the current obstacle is not allowed.

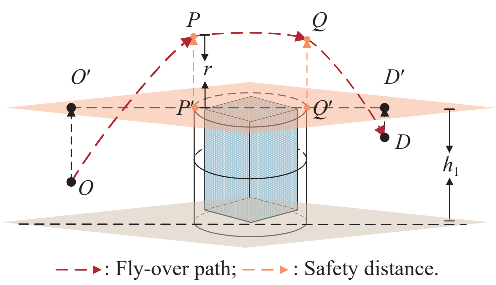


Fig. 7 Selection of fly over points

(iii) Selection of optimal waypoints

Once the fly-over and fly-around points have been generated, the way in which the UAV bypasses the obstacle is determined according to an optimal waypoint selection strategy.

The final waypoint is selected from the two ways of flying around or flying over, based on the number of existing obstacles on the sub-path and the energy consumption. It is noteworthy that flying over a building may sometimes lead to greater energy consumption. However, maintaining a fly-over height can enable the UAV to bypass multiple obstacles directly in subsequent paths, thereby eliminating the need to shuttle between different altitudes. Consequently, we take into account the height of the subsequent obstacles to predict total energy consumption, aiding in the selection of the optimal waypoint. The pseudo-code of optimal waypoint selection algorithm is shown as Algorithm 3.

Algorithm 3 Waypoint selection algorithm

Input O, D, optimal waypoint for fly-around L, fly over points P, Q, fixed cruising altitude H
Output Waypoint
Calculate the obstacles that conflict with the paths OL, OP, LD, QD, denoted as $C_{\text{OL}}, C_{\text{OP}}, C_{\text{LD}}, C_{\text{QD}}$ respectively.
If $h_P > H$:
  Return L
Else
  If $h_I + r > H$ for I in C_{QD}:
    If $E(\text{OLD}) \leqslant E(\text{OPQD})$:
      Return L
    Else
      Return P, Q
  Else
    Continue
  If $\text{num}(C_{\text{OL}}) + \text{num}(C_{\text{LD}}) = \text{num}(C_{\text{OP}}) + \text{num}(C_{\text{QD}})$:
    If $E(\text{OLD}) \leqslant E(\text{OPQD})$:
      Return L
    Else
      Return P, Q
  Elif $\text{num}(C_{\text{OL}}) + \text{num}(C_{\text{LD}}) > \text{num}(C_{\text{OP}}) + \text{num}(C_{\text{QD}})$:
    Return P, Q
  Else
    Return L

Firstly, if the height of the obstacle exceeds the maximum allowable height, the optimal choice is to fly-around it. However, if there is a subsequent obstacle whose height is greater than the maximum height, the option with the lowest energy consumption should be selected. In scenarios where none of the subsequent obstacles exceed the pre-established maximum flight height, the decision should be predicated on the quantity of conflicting obstacles. In cases of equivalency, the selection should once more favour the path that minimises energy consumption.

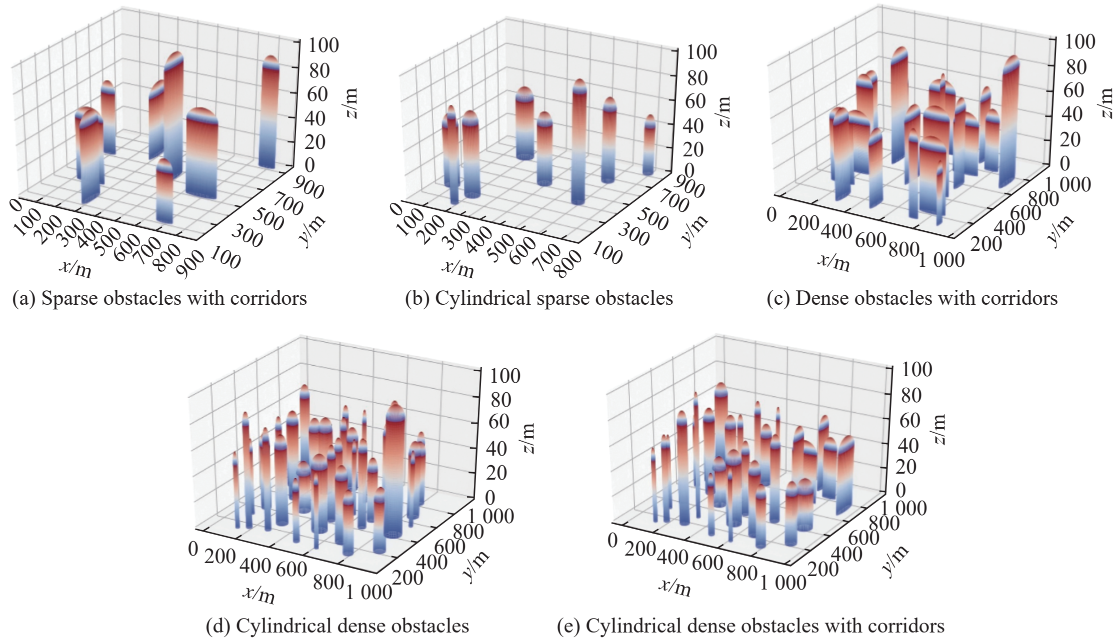
5. Experiments and results

In this section, experiments based on randomly generated examples are conducted to test the effectiveness of the proposed algorithms. Furthermore, the sensitivity analysis of some factors, including UAV cruising altitude and safety distance, is carried out with a real case.

5.1 Experiment on random instances

5.1.1 Experiment design

Fifteen examples are randomly generated based on five different environments [55], and the obstacles in different environments have different densities and shapes. The five kinds of obstacle distribution are shown in Fig. 8, namely, sparse obstacles with corridors, cylindrical sparse obstacles, dense obstacles with corridors, cylindrical dense obstacles, and cylindrical dense obstacles with corridors.



(a) Sparse obstacles with corridors (b) Cylindrical sparse obstacles (c) Dense obstacles with corridors

(d) Cylindrical dense obstacles (e) Cylindrical dense obstacles with corridors

Fig. 8 Example instances in five environments

Specifically, details of the 15 instances are shown in Table 3. The first five examples (E1−E5) generate simulated scenarios with a range of 1 km×1 km×0.1 km, while the other examples (E6−E15) generate simulated scenarios with a range of 2 km×2 km×0.1 km. The last five examples (E11−E15) further increase the number of obstacles in addition to extending the map range. The height of all obstacles is randomly generated in the range of 30−100 m.

Table 3 Details of the fifteen examples

Example	Environment	Number of obstacles	Size/km	Height of obstacles/m
E1	Fig. 8(a)	8	1*1	30−100
E2	Fig. 8(b)	8	1*1	30−100
E3	Fig. 8(c)	20	1*1	30−100
E4	Fig. 8(d)	40	1*1	30−100
E5	Fig. 8(e)	30	1*1	30−100
E6	Fig. 8(a)	8	2*2	30−100
E7	Fig. 8(b)	8	2*2	30−100
E8	Fig. 8(c)	20	2*2	30−100
E9	Fig. 8(d)	40	2*2	30−100
E10	Fig. 8(e)	30	2*2	30−100
E11	Fig. 8(a)	32	2*2	30−100
E12	Fig. 8(b)	32	2*2	30−100
E13	Fig. 8(c)	80	2*2	30−100
E14	Fig. 8(d)	160	2*2	30−100
E15	Fig. 8(e)	120	2*2	30−100

In terms of UAV parameters, the minimum turning radius is set at 10 m for both horizontal and vertical directions, the cruising altitude at 60 m and the maximum flight altitude at 90 m. When calculating the energy consumption of the UAV, different avoidance methods lead to different energy consumption rates, and fly-over consumes more energy than fly-around in the same time. The UAV parameters are referred to [48] and the maximum total energy consumption of the UAV is set to 1 500 Wh.

5.1.2 Experiment result

Four algorithms, Dubins-RRT* [56], SAS [27], APF [57], and 3D-target guidance (3D-TG) [32], are selected as comparison algorithms to verify the superiority of the proposed 3DE-PP. Detailed calculation results are shown in Table 4 and Fig. 9, including the energy consumption and the running time.

Table 4 Experimental results in simulated environment

Example	Energy consumption/kWh					Gap/%					Time/s				
	Dubins-RRT*	SAS	APF	3D-TG	3DE-PP	Dubins-RRT*	SAS	APF	3D-TG	3DE-PP	Dubins-RRT*	SAS	APF	3D-TG	3DE-PP
E1	1.73	1.58	1.60	1.60	1.55	11.54	2.26	3.59	3.12	0.00	343.23	0.33	0.01	0.02	1.03
E2	1.60	1.57	1.57	1.59	1.53	4.75	2.90	2.99	4.02	0.00	539.25	0.56	0.01	0.00	0.91
E3	1.67	1.59	1.76	1.61	1.57	6.95	1.35	12.76	2.86	0.00	525.46	0.60	0.02	0.01	1.26
E4	1.85	1.57	1.87	1.56	1.61	18.76	1.11	20.47	0.00	3.48	556.02	0.02	0.02	0.01	1.11
E5	1.89	1.56	2.46	1.58	1.54	22.47	1.02	59.56	2.71	0.00	491.28	0.01	0.03	0.01	1.24
E6	3.10	3.09	2.35	2.82	2.72	31.52	31.18	0.00	19.87	15.56	693.71	0.09	0.01	0.44	0.96
E7	3.18	2.89	2.77	2.83	2.74	16.19	5.83	1.13	3.54	0.00	490.25	0.96	0.04	0.01	0.92
E8	3.33	2.88	3.02	2.84	2.74	21.64	5.21	10.33	3.72	0.00	509.06	1.09	0.03	0.02	1.17
E9	3.56	3.22	3.15	2.92	2.90	22.72	11.19	8.67	0.89	0.00	305.78	1.26	0.02	0.03	1.54
E10	2.84	2.92	3.30	2.84	2.81	1.16	3.95	17.65	1.27	0.00	775.14	1.84	0.07	0.01	1.71
E11	3.79	2.96	2.92	2.78	2.77	36.97	6.89	5.63	0.56	0.00	402.80	0.07	0.05	0.01	1.31
E12	3.33	2.92	3.05	2.88	2.81	18.37	3.76	8.23	2.33	0.00	661.65	0.05	0.02	0.01	0.88
E13	3.59	2.92	3.50	2.83	2.89	27.01	3.20	23.73	0.00	2.14	920.22	0.53	0.04	0.03	1.76
E14	3.94	2.81	3.72	2.91	2.87	40.21	0.00	32.28	3.40	2.12	1017.64	0.24	0.10	0.03	1.74
E15	3.53	2.81	3.27	2.82	2.80	26.26	0.25	16.83	0.76	0.00	727.30	0.91	0.07	0.02	1.63

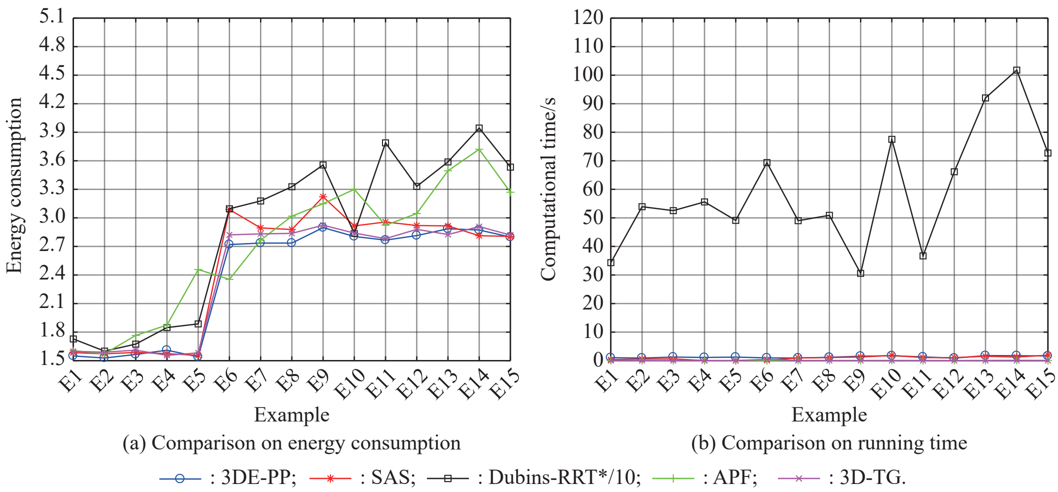


Fig. 9 Comparison of five algorithms

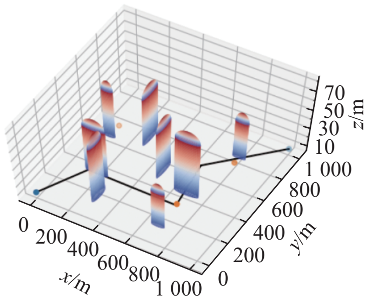
As evidenced by Table 4, the designed 3DE-PP algorithm generates better paths than the other four algorithms on a larger number of cases. Calculating the difference from the optimal solution by gap = solution−optimal/optimal, the average difference from the optimal solution for the Dubins-RRT* algorithm is about 20.4%, while the average difference from the optimal solution for the SAS algorithm is about 5.3%, for the APF algorithm it is about 14.9%, and for the 3D-TG algorithm and our algorithm the average difference from the optimal solution is 3.2% and 1.5%, respectively. In comparison, our algorithm outperforms the other four algorithms overall. Compared to Dubins-RRT*, which generates waypoints by random sampling, our algorithm outperforms it significantly in terms of solution effectiveness and speed. We have also noted that the average disparity between the Dubins-RRT* algorithm and the 3DE-PP algorithm exhibits an ascending trend, with the differences being approximately 12% for E1−E5, 18% for E6−E10, and 30% for E11−E15. This can be attributed to the increasing size and complexity of the environment, which subsequently enlarges the search space and compounds the difficulty of achieving superior solutions through random sampling.

The SAS algorithm generates a degraded solution quality in the corridor environment and its solution time gradually rises with the density of obstacles. The APF algorithm performs well in scenarios with sparse obstacles but begins to deteriorate as obstacles become denser, which is due to the strong repulsive forces that may lead to the oscillation of the UAV in the corridor environment. The 3DE-PP algorithm, on the other hand, focuses only on obstacles in the direction of flight and does not search the whole area, thus searching faster and getting better solutions.

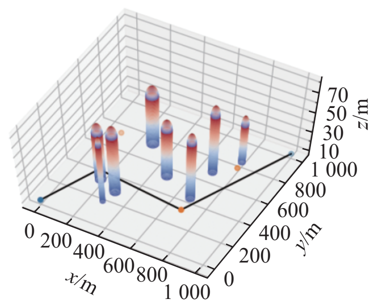
Fig. 9 shows the trend of energy consumption and computation time for the five algorithms. It is evident that the other four algorithms remain relatively unaffected by the various cases. However, the Dubins-RRT* algorithm exhibits a significant increase in computation time, which is due to the fact that the algorithm searches for a feasible solution through a random search. The process becomes increasingly complex and time-consuming as the size of the region and the density of obstacles grow.

Overall, 3DE-PP outperforms the other methods in most cases, and in terms of solution time, the last four algorithms are all controlled to be around 1 s, except for the Dubins-RRT* algorithm.

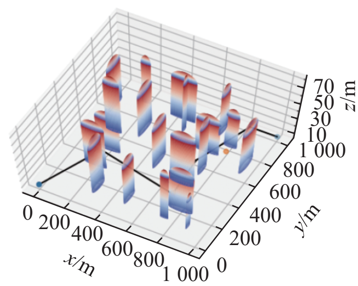
Fig. 10 shows the paths obtained through 3DE-PP for the 15 cases. As can be seen from the figure, as the map size and the number of obstacles increase, the UAV prefers to fly over the obstacles, which saves more energy and the total path deviates less from the original straight path. It can also be seen from the figure that our algorithm can jump out of the maze environment because it can fly over buildings. In addition, the UAV path hardly circles, which shows that our angular algorithm performs well in reducing circling during takeoff and landing.



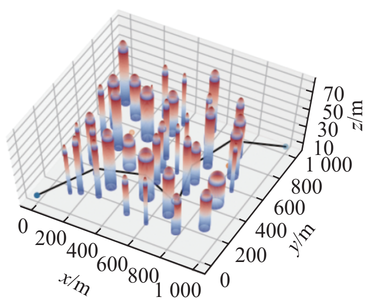
(a) Result of E1



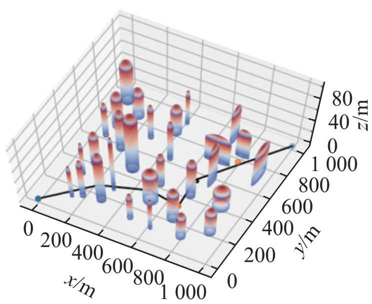
(b) Result of E2



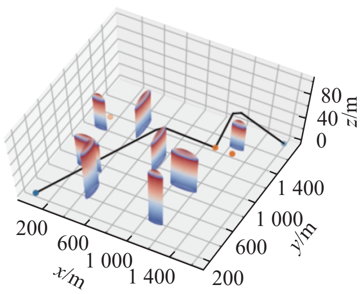
(c) Result of E3



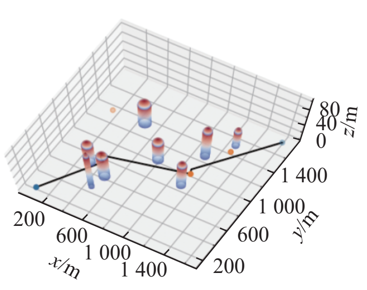
(d) Result of E4



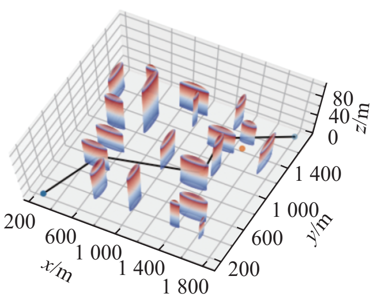
(e) Result of E5



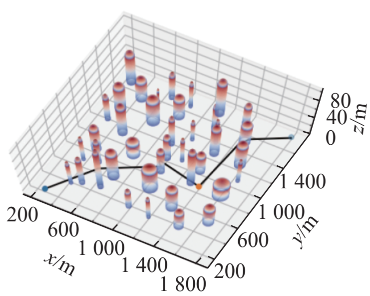
(f) Result of E6



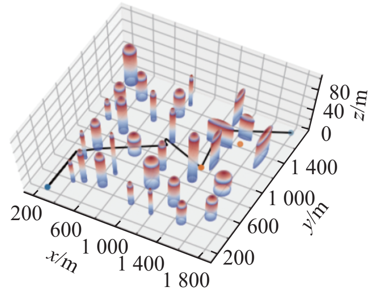
(g) Result of E7



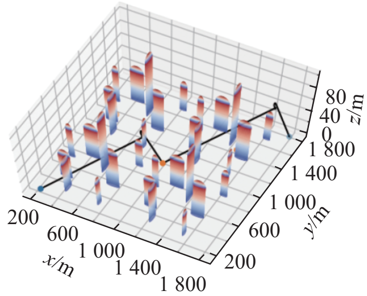
(h) Result of E8



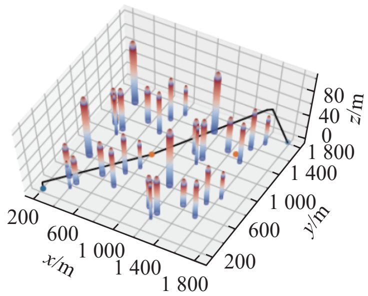
(i) Result of E9



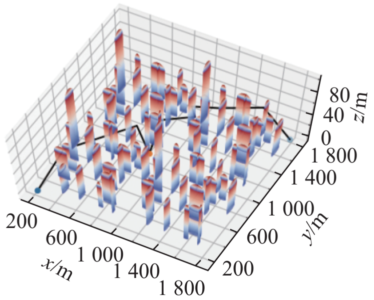
(j) Result of E10



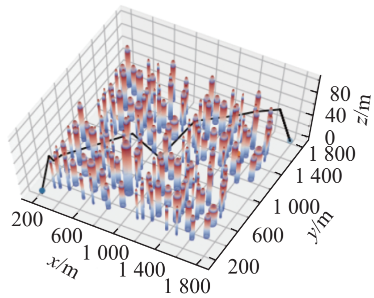
(k) Result of E11



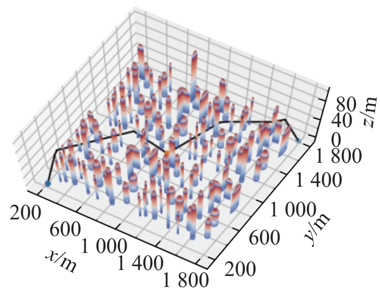
(l) Result of E12



(m) Result of E13



(n) Result of E14



(o) Result of E15

• : Customers; • : Charging stations.

Fig. 10 Presentation of the path of the examples

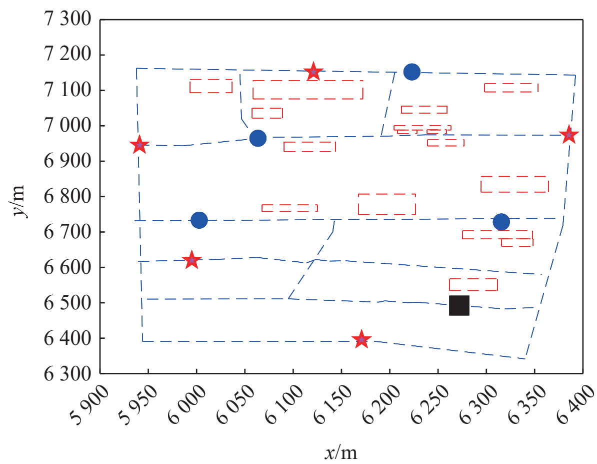
In summary, the proposed 3DE-PP not only provides flyable safe paths that satisfy the constraints of UAV motion and cruise altitude, but also pays attention to the obstacles in the direction of UAV motion, avoiding ineffective computation and improving the solution speed. Moreover, by optimising the flight angle, the path generated by 3DE-PP is able to navigate closer to obstacles, effectively avoiding unnecessary detours.

5.2 Case study

5.2.1 Case description

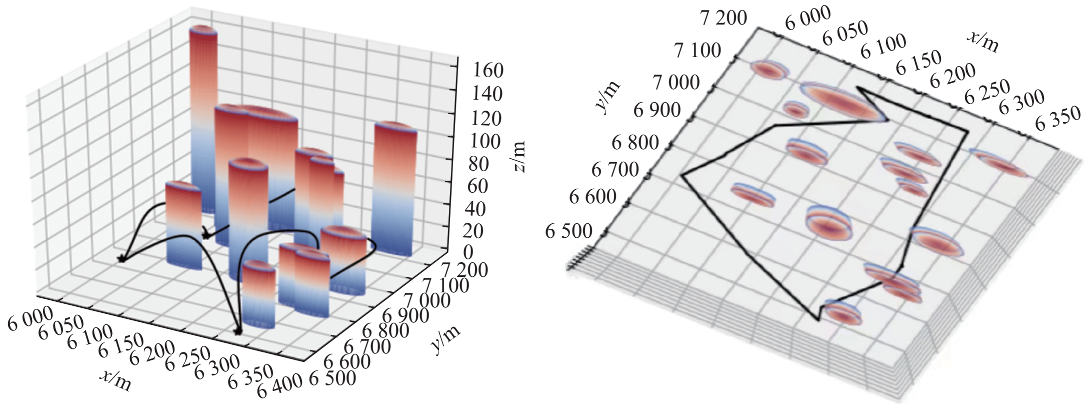
In this subsection, a real case study is built based on the distribution of buildings. As depicted in Fig. 11, an urban area is selected as the experimental space, encompassing 14 buildings with heights ranging from 50 m to 200 m. These buildings are potential obstacles in the UAV flight path and are modeled to generate smooth geometries, as illustrated in Fig. 13(a). Within this area, five customer locations and five charging stations are strategically distributed. The spatial arrangement of these obstacles and corresponding points is presented in Fig. 12. Fig. 12 customer locations are denoted by blue dots, charging stations are indicated by red pentagrams, and the depot is represented by a black square.

Fig. 11 Layout of the actual case



■ : Deport; ● : Customers; ★ : Charging station.

Fig. 12 Distribution of customers and charging station



(a) 3D result of the actual case

(b) 2D result of the actual case

Fig. 13 Results of the actual case

The minimum turning radius and climb rate of the UAV are set to 1 m according to [58], the safety distance is set to 5 m, the cruise altitude is 60 m and the maximum flight altitude is set to 150 m.

The 3DE-PP algorithm has been implemented to address the given scenario, with the outcomes depicted in Fig. 13. Where Fig. 13(a) presents the 3D situation and Fig. 13(b) displays an overhead view. The cumulative energy consumption at cruising altitude of 80 is 1 766.437 units, with the computation being completed within 2 s. It can be observed from Fig. 13(a) that the UAV would fly over lower obstacles while fly around higher buildings. From Fig. 13(b), the UAV flies over two obstacles and around the three obstacles.

5.2.2 Sensitivity analysis

Two factors are considered in the sensitivity test, which are the safety distance and the cruising altitude.

(i) Impact of the safety distance

Considering the safety of the UAV flying around obstacles, a safety distance is set. To examine the effect of the safety distance on the energy consumption, the algorithm considering only fly-around is used as comparison while the safety distance changes from 1 to 15. The results are presented in Table 5 and Fig. 14. As displayed in Table 5, with the increase of the safety distance, the path energy consumption of the 3DE-PP algorithm increased by 8.7% without fly-over, while the same algorithm only led to a 3.5% increase in path energy consumption. This demonstrates that our algorithm is both stable and insensitive to the safety distance.

Table 5 Impact of safety distances

Example	Energy consumption/Wh	
	3DE-PP without fly-over	3DE-PP
1	1 807.586	1 807.586
2	1 812.672	1 812.672
3	1 843.703	1 844.772
4	1 848.161	1 849.374
5	1 765.162	1 766.437
6	1 829.947	1 871.858
7	1 820.236	1 844.217
8	1 788.739	1 857.109
9	1 830.160	1 849.126
10	1 818.768	1 837.652
11	1 825.433	1 873.088
12	1 873.707	1 922.271
13	1 877.794	1 942.789
14	1 879.739	1 942.802
15	1 872.559	1 965.317

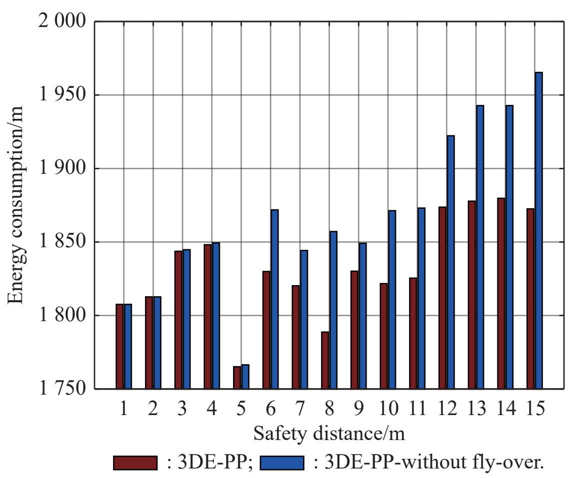


Fig. 14 Computational results under different safety distance

As can be seen from Fig. 14, energy consumption presents an overall upward trend rather than a monotonic steady increase, since Dubins curves are adopted to generate flight paths between waypoints. At different flight heights and safe distances, differences in surrounding obstacles, distance to obstacles and flight angles can lead to differences in Dubins paths.

(ii) Impact of the cruising altitude

In addition to the safety distance, the influence of cruising altitude on energy consumption was also examined. For comparative purposes, the algorithm that solely considers the fly-around strategy is utilized. The impact on energy consumption was analyzed as the cruising altitude varied from 50 m to 190 m. The corresponding results are presented in Table 6 and Fig. 15.

Table 6 Impact of cruising altitude

Cruising altitude	Energy consumption/Wh	
	3DE-PP without fly-over	3DE-PP
50	1749.775	1747.953
60	1774.064	1778.995
70	1794.236	1806.918
80	1818.768	1837.652
90	1842.041	1864.585
100	1872.544	1900.014
110	1916.557	1937.413
120	1962.366	1976.576
130	2003.496	2023.602
140	2046.078	2065.774
150	2089.969	2102.962
160	2147.589	2147.589
170	2205.839	2205.839
180	2239.936	2239.936
190	2287.467	2287.467

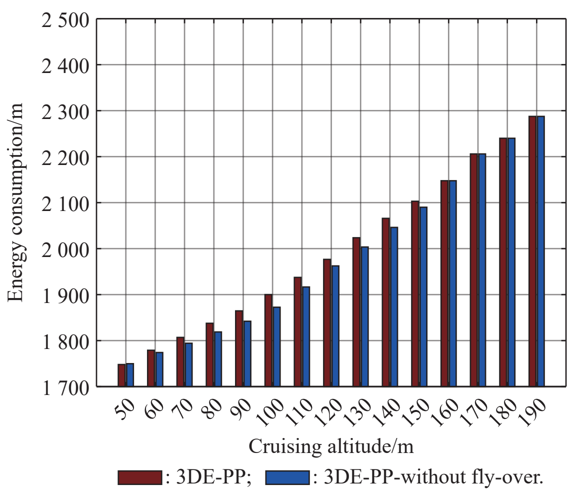


Fig. 15 Computational results under different cruising altitudes

It can be seen that when the cruise altitude is 50 m, the difference in energy consumption is small, when the cruise altitude is 100 m, the difference in energy consumption increases significantly, and when the cruise altitude exceeds 150 m, the energy consumption tends to be the same. It can be inferred that when the cruise altitude is low, the difference between the obstacle height and the cruise altitude is large, and flying over the obstacle will lead to larger energy consumption, so flying around is a better choice. As the cruise height increases, it is more preferred to fly over the obstacles, and finally when the cruise height is higher than the buildings, there is no conflict, so the energy consumption of both algorithms is the same. The results show that our algorithms can be optimal when choosing between different obstacle avoidance methods such as fly-around or fly-over.

As illustrated in Fig. 15, which is because there is a shift in the obstacle avoidance mode and therefore the difference increases. However, the energy consumption of the UAV does not increase steadily with the increase of cruising altitude. It does not increase after reaching 150 m, this is because the maximum height of our obstacles is set to 150 m, therefore, there is no obstacle above this cruising altitude and the flight path is consistent.

6. Conclusions

In order to effectively plan smooth and safe paths for UAVs in complex 3D environments and consider UAV energy limitations, this paper proposes a 3D envelope-based path planning method which takes the UAV’s turn radius, maximum flight height, maximum climb rate and UAV energy limitations into account. The algorithm first generates the initial route by considering UAV energy limitations through an improved saving algorithm. Secondly, elliptical cylinders and ellipsoidal envelopes of the obstacles are constructed to smooth them out. A 3D path between the target points is then generated based on the improved Dubins algorithm. If the path conflicts with an obstacle, construct its fly-around and fly-over waypoints based on tangent guidance strategies and minimum deviation-based strategies, respectively. If the path conflicts with an obstacle, its bypass and flyover path points are constructed based on a tangent guidance strategy and a minimum deviation-based strategy, respectively. The best waypoints are determined by defined rules and Dubins paths are constructed between these points. Recycle the above process until a collision-free path is generated.

Random experiments are conducted in different environments and the results show the effectiveness of the algorithm. The algorithm can always find the path with the least consumption. In addition, safety distances and cruising altitude are considered in sensitivity analysis based on a real case. The experimental results show that the algorithm tends to fly over the obstacles as the safety distance increases, while it tends to fly around obstacles as the cruising altitude decreases.

Although the algorithm in this paper can generate high-quality collision-free paths, it does so under the condition that obstacles are known, and obstacles are not completely known in UAV online planning, so our next research plan is to extend the above work to environments consisting of dynamic obstacles to better simulate real-world scenarios.

References

[1] BURNS L D. A vision of our transport future. Nature, 2013, 497(7448): 181–182.

[2] LE V S, ZOLFAGHARI A, POLAK J. Autonomous cars: the tension between occupant experience and intersection capacity. Transportation Research Part C: Emerging Technologies, 2015, 52: 1–14.

[3] JARAŠŪNIENĖ A, IŠORAITĖ M, PETRAŠKA A. Developing small-cargo flows in cities using unmanned aerial vehicles. Future Transportation, 2024, 4(2): 450–474.

[4] FANG Z, SAVKIN A V. Strategies for optimized UAV surveillance in various tasks and scenarios: a review. Drones, 2024, 8(5): 193–233.

[5] AGRAWAL J, ARAFAT M Y. Transforming farming: a review of AI-powered UAV technologies in precision agriculture. Drones, 2024, 8(11): 664–688.

[6] SUN C, FONTANESI G, CANBERK B, et al. Advancing UAV communications: a comprehensive survey of cutting-edge machine learning techniques. IEEE Open Journal of Vehicular Technology, 2024, 5: 825–854.

[7] PAN Y, LI L F, QIN J J, et al. Unmanned aerial vehicle-human collaboration route planning for intelligent infrastructure inspection. Computer-Aided Civil and Infrastructure Engineering, 2024, 39(14): 2074–2104.

[8] ZHANG J, DU X, DONG Q C, et al. Distributed collaborative complete coverage path planning based on hybrid strategy. Journal of Systems Engineering and Electronics, 2023, 35(2): 463–472.

[9] YANG L M, ZHANG X Y, LI Z P, et al. A LODBO algorithm for multi-UAV search and rescue path planning in disaster areas. Chinese Journal of Aeronautics, 2025, 38(2): 1000–9361.

[10] JUNG S, KIM H. Analysis of amazon prime air UAV delivery service. Journal of Knowledge Information Technology and Systems, 2017, 12(2): 253–266.

[11] LEVIN A. Alphabet and chipotle are bringing burrito delivery drones to campus. https://www.bloomberg.com/news/articles/2016-09-08/burrito-by-drone-coming-to-campus-in-test-of-alphabet-s-delivery.

[12] FAN F F. JD. com opens first unmanned sorting center. https://www.chinadaily.com.cn/specials/0803huawei.pdf.

[13] MOHANTY S N, RAVINDRA J V R, NARAYANA G S, et al. Drone technology: future trends and practical applications. Hoboken: John Wiley & Sons, 2023.

[14] WALKER R, JIWANI R. Reinventing e-commerce: Amazon's bet on unmanned vehicle delivery. London: Kellogg School of Management, 2015.

[15] KELLEHER K. UPS, matternet delivering medical samples in North Carolina using drones. https://www.auvsi.org/industry-news/ups-matternet-deliveringmedical-samples-north-carolina-using-drones.

[16] LEE Y M. A study on the current status and case of the logistics industry using drone. Journal of Korea Culture Industry, 2020, 20(4): 75–85.

[17] YANG L, QI J, XIAO J, et al. A literature review of UAV 3D path planning. Proc. of the 11th World Congress on Intelligent Control and Automation, 2014: 2376−2381.

[18] MAZAHERI H, GOLI S, NOUROLLAH A. A survey of 3D space path-planning methods and algorithms. ACM Computing Surveys, 2024, 57(1): 1–32.

[19] KHAN M T R, MUHAMMAD S M, RU Y, et al. Aspects of unmanned aerial vehicles path planning: overview and applications. International Journal of Communication Systems, 2021, 34(10): 1074–5351.

[20] CHEN Q, WANG Y J, SUN Y F. An improved dung beetle optimizer for UAV 3D path planning. The Journal of Supercomputing, 2024, 80(18): 26537–26567.

[21] LUO J H, TIAN Y X, WANG Z Y. Research on unmanned aerial vehicle path planning. Drones, 2024, 8(2): 51.

[22] PHALAPANYAKOON K, SIRIPONGWUTIKORN P. Route planning of heterogeneous unmanned aerial vehicles under recharging and mission time with carrying payload constraints. Journal of Industrial Engineering and Management, 2023, 16(2): 215–235.

[23] CHEN F, ZHENG Y, WANG Z, et al. RBI-RRT*: efficient sampling-based path planning for high-dimensional state space. Proc. of the IEEE International Conference on Robotics and Automation. 2024: 8721−8727.

[24] LIU M J, ZHANG H X, YANG J, et al. A path planning algorithm for three-dimensional collision avoidance based on potential field and B-spline boundary curve. Aerospace Science and Technology, 2024, 144: 1270–9638.

[25] JIANG Y, XU X X, ZHENG M Y, et al. Evolutionary computation for unmanned aerial vehicle path planning: a survey. Artificial Intelligence Review, 2024, 57: 267.

[26] ZHANG J D, GUO Y K, ZHENG L H, et al. Real-time UAV path planning based on LSTM network. Journal of Systems Engineering and Electronics, 2024, 35(2): 374–385.

[27] SZCZERBA R J, GALKOWSKI P, GLICKTEIN I S, et al. Robust algorithm for real-time route planning. IEEE Trans. on Aerospace and Electronic Systems, 2000, 36(3): 869–878.

[28] LIU S, WATTERSON M, MOHTA K, et al. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. IEEE Robotics and Automation Letters, 2017, 2(3): 1688–1695.

[29] ZHANG N, ZHANG M, LOW K H. 3D path planning and real-time collision resolution of multirotor drone operations in complex urban low-altitude airspace. Transportation Research Part C: Emerging Technologies, 2021, 129: 103123.

[30] ROHNERT H. Shortest paths in the plane with convex polygonal obstacles. Information Processing Letters, 1986, 23(2): 71–76.

[31] LIU Y H, ARIMOTO S. Path planning using a tangent graph for mobile robots among polygonal and curved obstacles: communication. The International Journal of Robotics Research, 1992, 11(4): 376–382.

[32] LIU H, WU G H, ZHOU L, et al. Tangent-based path planning for uav in a 3-D low altitude urban environment. IEEE

Trans. on Intelligent Transportation Systems, 2023, 24(11): 12062–12077.

[33] CHEN H, CHANG K, AGATE C S. UAV path planning with tangent-plus-Lyapunov vector field guidance and obstacle avoidance. IEEE Trans. on Aerospace and Electronic Systems, 2013, 49(2): 840–856.

[34] LIU Y S, WANG Q X, HU H S, et al. A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes. IEEE Trans. on Systems, Man, and Cybernetics: Systems, 2018, 49(11): 2362–2372.

[35] LIU H, LI X M, FAN M F, et al. An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy. IEEE Trans. on Intelligent Transportation Systems, 2020, 23(4): 3061–3073.

[36] YANG F, LEE D H, KELLER J, et al. Graph-based topological exploration planning in large-scale 3D environments. Proc. of the IEEE International Conference on Robotics and Automation, 2021: 12730−12736.

[37] LIANG X, MENG G L, XU Y M, et al. A geometrical path planning method for unmanned aerial vehicle in 2D/3D complex environment. Intelligent Service Robotics, 2018, 11(3): 301–312.

[38] KARAMAN S, WALTER M R, PEREZ A, et al. Anytime motion planning using the RRT. Proc. of the IEEE International Conference on Robotics and Automation. 2011: 1478−1483.

[39] LIU J, XU T, YANG S X, et al. Navigation and visual feedback control for magnetically driven helical miniature swimmers. IEEE Trans. on Industrial Informatics, 2019, 16(1): 477–487.

[40] GAMMELL J D, SRINIVASA S S, BARFOOT T D. Informed RRT*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014: 2997−3004.

[41] WARREN C W. Global path planning using artificial potential fields. Proc. of the IEEE International Conference on Robotics and Automation, 1989: 316−321.

[42] ROBERGE V, TARBOUCHI M, LABONTÉ G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. on Industrial Informatics, 2012, 9(1): 132–141.

[43] WU Y, LOW K H, PANG B, et al. Swarm-based 4D path planning for drone operations in urban environments. IEEE Trans. on Vehicular Technology, 2021, 70(8): 7464–7479.

[44] VENTURINI F, MASON F, PASE F, et al. Distributed reinforcement learning for flexible UAV swarm control with transfer learning capabilities. Proc. of the 6th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, 2020. DOI: 10.1145/3396864.3399701.

[45] ZHANG C Q, ZHOU W J, QIN W D, et al. A novel UAV path planning approach: heuristic crossing search and rescue optimization algorithm. Expert Systems with Applications, 2023, 215: 119–243.

[46] XU J Y, SHI M J, TANG F, et al. Dubins-A*: a new global path planning scheme for fixed-wing UAV with irregular obstacles avoidance. Proc. of the 7th International Conference on Electronics Technology, 2024: 636−641.

[47] KUMAR D P, DARBHA S, MANYAM S G, et al. A new approach to motion planning in 3D for a dubins vehicle: special case on a sphere. https://arxiv.org/abs/2504.01215.

[48] FIGLIOZZI M A. Lifecycle modeling and assessment of unmanned aerial vehicles (drones) CO2e emissions. Transportation Research Part D: Transport and Environment, 2017, 57: 251–261.

[49] REKLEITIS I M, DUDEK G, MILIOS E E. Graph-based exploration using multiple robots. Proc. of the International Symposium on Distributed Autonomous Robotic Systems, 2000: 241−250.

[50] CLARKE G, WRIGHT J W. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 1964, 12(4): 568–581.

[51] CHITSAZ H, LAVALLE S M. Time-optimal paths for a Dubins airplane. Proc. of the IEEE 46th Conference on Decision and Control, 2007: 2379−2384.

[52] MCLAIN T, BEARD R W, OWEN M. Implementing Dubins airplane paths on fixed-wing UAVs. Dordrecht: Springer Netherlands, 2014.

[53] HOTA S, GHOSE D. Optimal geometrical path in 3D with curvature constraint. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010: 113−118.

[54] FREITAS E J R, COHEN M W, NETO A A, et al. DE3D-NURBS: a differential evolution-based 3D path-planner integrating kinematic constraints and obstacle avoidance. Knowledge-Based Systems, 2024, 300: 112084.

[55] TSARDOULIAS E G, ILIAKOPOULOU A, KARGAKOS A, et al. A review of global path planning methods for occupancy grid maps regardless of obstacle density. Journal of Intelligent & Robotic Systems, 2016, 84(1): 829–858.

[56] BI C Y, LIU J H, WANG J A, et al. Motion planning for fixed-wing UAV using modified Dubins-RRT* algorithm. Proc. of the International Conference on Guidance, Navigation and Control, 2024: 128−138.

[57] LAU D, EDEN J, OETOMO D. Fluid motion planner for nonholonomic 3-D mobile robots with kinematic constraints. IEEE Trans. on Robotics, 2015, 31(6): 1537–1547.

[58] COHEN I, EPSTEIN C, SHIMA T. On the discretized dubins traveling salesman problem. IISE Transactions, 2017, 49(2): 238–254.