J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 7829--7861

Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

# Improved RRT* algorithm UAV path planning based on multi-strategy fusion

Yuanyao Zhang[1], Hongji Liang[1], Junli Li[1,✉]

[1] *Information Engineering and Automation School of Kunming University of Science and Technology, Kunming, Yunnan, 650500, China*

ABSTRACT

Aiming at the RRT* algorithm in unmanned aerial vehicle (UAV) path planning, there are problems such as poor target bias, slow convergence speed, and tortuous path. This paper introduces an improved Bi-Informed-RRT* algorithm (BPD-APF-Informed-FARRT*), integrating a dual-path balancing operation strategy, a partition-biased sampling strategy, an artificial potential field guidance approach, and a fuzzy adaptive step-size strategy. To begin, the third point between the start and target points is chosen as the middle point, allowing four random trees to be generated at the same time at the start, target, and middle points, hence resolving the delayed convergence issue. Second, the artificial potential field method and the partition-biased sampling strategies are employed in both path generation and optimization phases to guide the placement of new nodes, tackling issues with poor target bias. Then, to address the intricacies of global environments, a fuzzy adaptive step size adjustment strategy is incorporated to boost the exploration efficiency of the growing tree in complex obstacle scenarios. Finally, leveraging the principle of triangular inequality, redundant nodes are removed, and the path is refined using the B-spline curve. Path planning simulation experiments were performed using MATLAB software. The results show that BPD-APF-Informed-FARRT* has more significant advantages in many ways compared with the Bi-RRT*, Informed-RRT*, and Bi-Informed-RRT* algorithms. This improved algorithm is a practical and feasible method for solving similar problems.

*Keywords:* UAV, path planning, RRT *, partition-biased sampling, artificial potential field method

## 1. Introduction

With the in-depth research in the fields of modern computer technology, automatic control theory, new sensor technology, and artificial intelligence, the autonomous flight of UAVs has further developed explosively. UAVs are widely used in many fields such as search and rescue, intelligent

✉ Corresponding author.
*E-mail address:* a20222104028@stu.kust.edu.cn (J. Li).
Received 12 January 2024; Revised 15 March 2024; Accepted 20 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-429

agriculture, and data collection [1-3]. In these applications, autonomous flight has become the future development direction of UAVs, and the path planning and optimization problem of UAVs becomes crucial.

UAVs have static and dynamic obstacles in different application environments, and due to the unpredictability of external perturbations, UAVs face an exponential increase in complexity in sensing, decision-making, and control systems [4-5]. On the one hand, UAVs fly in harsh and diverse environments, and traditional sensing methods may not be able to correctly recognize obstacles in the surrounding environment, and the ability to predict the trajectory of dynamic obstacles is insufficient, which may lead to the inability of UAVs to correctly sense their surroundings and thus cause serious accidents [6-10]. On the other hand, UAVs need to have the ability to avoid dynamic obstacles in a stable, efficient and safe manner [11-12]. In addition to this, the operational safety and reliability of UAVs should not only comply with the basic motion planning routes, but also introduce some constraints and dynamics limitations to solve the problems of slow speed, stiff motion and limited flight continuity in autonomous UAV flight [13-16]. Efficient obstacle avoidance is the key to successful UAV mission execution in complex environments [17]. Obstacle avoidance relies on intelligent path planning, i.e., planning a collision-free path from the starting point to the destination point that satisfies certain constraints, which centers on path planning algorithms [18-19]. The rule-based dynamic obstacle avoidance control process for complex UAS has been proven to be computationally expensive, while the UAV obstacle avoidance algorithms themselves are less adaptive and self-learning [20-22]. Therefore, the proposed improved RRT* algorithm UAV path planning strategy based on multi-strategy fusion plays a key role in improving the operational capability of UAS in complex environments [23-24].

Currently, commonly used path planning algorithms mainly include traditional algorithms and swarm intelligence algorithms. The traditional algorithms include Dijkstra's algorithm, A* algorithm, Probabilistic Roadmap (PRM), Rapidly Expanded Random Tree (RRT), and Artificial Potential Field Method. Literature [25] uses matrix-aligned Dijkstra's algorithm to plan optimal routes for UAV trajectories by modeling the dynamic grid environment and finding the safest path through high-dimensional matrices for unexplored regions. Literature [26] combines UAVs and spatial regions to establish a spatial model based on mixed integer linear programming, and generates the optimal flight path for UAVs using the improved A* algorithm based on the improved evaluation function and node selection strategy. However, the above two algorithms mainly address the time and partial accuracy of route planning, but both are time consuming. Literature [27] designed a dynamic exploration planner equipped with incremental sampling and probabilistic roadmap to plan the UAV's flight path, gradually increasing the sampling nodes and distributing them uniformly in the exploration area, which improves the UAV's dynamic obstacle avoidance capability. The sampling-based PRM algorithm, however, has a high computational cost and is very dependent on the number and quality of sampling points. Literature [28] proposed the artificial potential function to solve the UAV trajectory planning and formation control problem under 3D constraints, which can effectively calculate the optimal path of the UAV from the initial position to the target position and ensure the existence of the desired angle and distance between the UAVs. Although the artificial potential field method is real-time and simple to compute, it is prone to local minima and oscillations. For the RRT algorithm, literature [29] also shows that the RRT algorithm is suitable for path planning of autonomous mobile robots, but the high randomness of the sampling process, the low sampling efficiency, and the difficulty of finding the optimal path solution are the significant drawbacks of the RRT path planning algorithm. And swarm intelligence algorithms include ant colony algorithm, genetic algorithm, sparrow algorithm and so on. Literature [30] proposed UAV obstacle avoidance path planning based on multiple swarm ant colony algorithms for the problem of suboptimal solution convergence of a single ant colony optimization algorithm, and cooperated to find the optimal solution through information exchange between ant colonies to achieve the effectiveness of trajectory planning. Literature [31] explored the emergency landing path planning method for UAVs in different scenarios, compared with other algorithms, genetic algorithm can quickly calculate the optimal path solution with better quality under the premise of satisfying the problem constraints, and maximize the protection of personnel and property safety. Literature [32] evaluates the performance of chaotic sparrow search algorithm for planning UAV paths in three-dimensional complex environments, and the results show that the improved chaotic sparrow search algorithm, which introduces a nonlinear dynamic weighting factor and a dynamic boundary lens imaging inverse learning strategy, still has a high convergence efficiency in complex environments. Although swarm intelligence algorithms are more widely used in solving path planning problems, the above algorithms generally have the problems of slow convergence speed, long time in planning, and easy to fall into local optimum.

The RRT* algorithm can make the UAV navigation path tend to be optimal by adding the neighborhood search and rewiring process, and a large number of scholars have carried out further research to address the shortcomings of the RRT* algorithm in the field of UAV path planning. Literature [33] proposed a two-way APF-RRT* algorithm for UAV trajectory planning, based on the two-way RRT* algorithm to generate random growth trees that can alternate with each other, and combined with the improvement of the artificial potential field method to reduce the number of iterations, in order to improve the search performance of the growth tree. Literature [34] based on the UAV flight process of the complex three-dimensional environment to propose the relevant flight constraints, and for this reason, the standard RRT * algorithm is extended, fast extended random tree * algorithm (FC-RRT *) planning at the optimal flight path significantly improves the path security and keep the distance is the shortest. Literature [35] proposes a data-driven feedback fast exploratory random tree stellar algorithm (FRRT*), which utilizes extractable information from the situational data to grow constraints on the random tree and performs biased adjustments, which greatly improves the efficiency of path planning. Literature [36] examined the application of the fast random tree (IRRT*) algorithm in UAV optimal path prediction, which combined with the triangular inequality reconnection technique to plan UAV collision-free trajectories in complex 3D environments, and was able to significantly improve the planning efficiency and reduce the costly distance. Literature [37] further combines the bidirectional artificial potential field method with the bidirectional bias sampling idea to construct the intelligent bidirectional RRT* (BPIB-RRT*) algorithm, which flexibly adjusts the sampling space on the basis of spatial exploration, greatly improves the sampling efficiency and convergence speed, and possesses a high superiority in path planning. Although the above algorithm improves the RRT* algorithm in terms of sampling method and expansion method, the improved algorithm still suffers from high search randomness, redundant paths and poor smoothing. Based on this, the improved RRT* algorithm based on multi-strategy fusion is proposed, aiming to achieve efficient optimization and planning of multi-UAV trajectories in complex nonlinear environments.

Inspired by the above literature, to solve the problems of the slow convergence speed of the algorithm, poor target orientation, and more redundant nodes. This paper introduces an improved Bi-Informed-RRT* algorithm, which integrates a dual-path balancing operation strategy, a partition-biased sampling strategy, an artificial potential field guidance strategy, and a fuzzy adaptive step-size strategy. These enhancements are specifically aimed at advancing UAV path planning methodologies. The main contributions of this paper are as follows:

1) The Bi-Informed-RRT* algorithm introduces a middle point, and four randomized trees are expanded in parallel between the initial and middle points and between the middle and target points. Four randomized trees are expanded simultaneously, significantly improving the algorithm's search capability.

2) The partition-biased sampling strategy is introduced in the initial path generation and path optimization process of the Bi-Informed-RRT* algorithm. The utilization of sampling points can be improved to obtain higher quality samples.

3) The improved artificial potential field method is fused with the Bi-Informed-RRT* algorithm to introduce the gravitational effect of the target point and the repulsive effect of the obstacles during the new node generation process to guide the location of the new node generation, increase the target pointing of the randomly sampled points, and reduce the path planning time.

4) By designing the fuzzy controller to adaptively adjust the step size to accelerate the avoidance of obstacle regions with different densities, the convergence speed of the algorithm is accelerated, and the quality of path planning is improved.

5) Yaw angle and pitch angle constraints are added to ensure the feasibility of the generated paths, the triangular inequality principle is adopted to remove the redundant nodes, and the paths are smoothed by the B-spline curve. The problem that the UAV has too large a turning angle in flight is solved.

## 2. Background knowledge

### 2.1. Problem definition

Give the relevant concepts and related definitions required in UAV path planning. In this paper, $X_{\text{free}} \subseteq R^n$ is defined as the state space of the planning problem, $X_{\text{obs}} \subseteq X$ is the collision space, $X_{\text{free}} = X \setminus X_{\text{obs}}$ is the free space, $X_{\text{star}} \in X_{\text{free}}$ is the initial state, and $X_{\text{goal}} \in X_{\text{free}}$ is the desired final state. $\sigma : [0,1] \to X_{\text{free}}$ is a feasible path and $\Sigma$ is the set of all feasible paths.

The optimal path planning problem, on the other hand, is a search for a path $\sigma^*$. The optimal path minimizes the cost function $c : \Sigma \to R^*$ when planning from the initial point $X_{\text{star}}$ to the target point $X_{\text{goal}}$ in the free space $X_{\text{free}}$, denoted as follows:

$$\begin{aligned} \sigma^* = \arg\ \min_{\sigma\in\Sigma} \{ & c(\sigma) \mid \sigma(0) = X_{\text{star}}, \\ & \sigma(1) = X_{\text{goal}}, \forall s \in [0,1], \sigma(s) \in X_{\text{free}} \} \end{aligned} \tag{1}$$

With its probabilistic completeness, the RRT algorithm ensures that a feasible path from the start to the target point can be found, but randomness leads to the fact that the obtained path may not be optimal. The RRT* algorithm optimizes this problem by introducing the mechanisms of reselecting the parent nodes and rewiring. Reselecting the parent nodes ensures that new nodes are added to the tree at the lowest cost, while rewiring streamlines the paths after each new node is generated, eliminating redundancy and reducing the path cost. These two processes work together to enable the RRT* algorithm to progressively approximate the optimal path.

### 2.2. UAV path planning constraints

During the UAV path planning process, ensuring that each newly added node is executable is crucial. Restrictions were implemented to maximize the steering angle and climb angle of the UAV. By setting a maximum steering angle, we limited the movement of the UAV in the horizontal plane and ensured that the yaw angle remained within safe limits when transitioning from one path point to the next. At the same time, the maximum climb angle limit is a constraint on the vertical movement of the UAV to ensure that the change in pitch angle is maintained within safe bounds as it ascends from one path point to the next. The limitations of yaw and pitch angles guarantee the dynamic safety and rationalization of the operation of the UAV when performing path planning.

Assume that the coordinates of the three neighboring path points during the UAV motion flight are given in the following equation:

$$\begin{cases} p_{i-1}(x_{i-1}, y_{i-1}, z_{i-1}) \\ p_i(x_i, y_i, z_i) \\ p_{i+1}(x_{i+1}, y_{i+1}, z_{i+1}) \end{cases} \tag{2}$$

The following discussion takes place:

1) Yaw Angle Constraint. In a three-dimensional coordinate system, the yaw angle $\psi$ of two adjacent path segments usually refers to the angle between the projections of these two paths on the horizontal plane. The yaw angle $\psi$ of two adjacent path segments is shown in Figure 1.

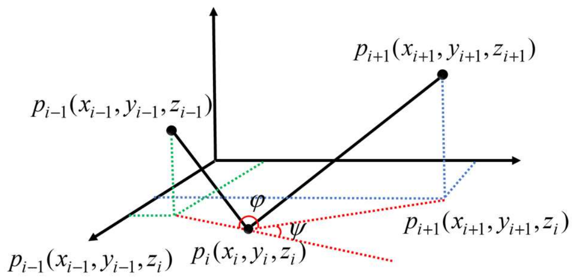



**Fig. 1.** Schematic of yaw angle.

The yaw angle $\psi$ should satisfy the constraint: $0 \le \psi \le \psi_{\max}$ . where $\psi_{\max}$ is the maximum yaw angle, $\varphi$ is the complementary angle of $\psi$ . Calculated according to the cosine theorem:

$$\cos\varphi = \frac{A+B+C}{2\sqrt{AB}} \tag{3}$$

$$\psi = -\arccos(\cos\varphi) \tag{4}$$

$A$ , $B$ and $C$ in equation (3) are denoted as:

$$\begin{cases} A = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \\ B = (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \\ C = (x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2 \end{cases} \tag{5}$$

2) Pitch angle constraints. The pitch angle $\theta$should satisfy constraint: $0 \le \theta \le \theta_{\max}$ , where $u$ is the maximum pitch angle, The pitch angle is shown as $\theta$in Figure 2.

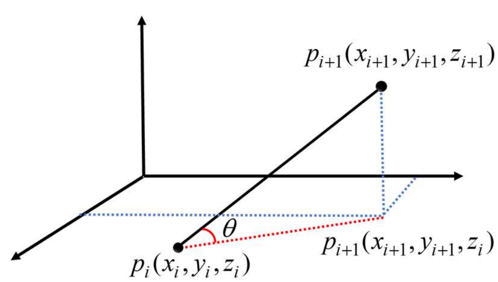



**Fig. 2.** Schematic of pitch angle.

Calculation gives:

$$\theta = \arctan\left(\frac{|z_{i+1} - z_i|}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}\right) \tag{6}$$

# 3. Algorithm Introduction

## 3.1. RRT*

Before introducing the RRT* algorithm, the RRT algorithm is first described to set the stage for introducing the RRT* algorithm. The classical RRT algorithm was first proposed by Lavalle in 1998. The RRT search process is similar to the process of expanding a tree branch in all directions. It generates a randomized search tree $T$ in the workspace, with the starting point $X_{star}$ denoting the root node of the tree. Random sampling is used to generate the sampling point $X_{rand}$, after which addressing in the tree to find the nearest node $X_{nearest}$ in the tree to $X_{rand}$, and expanding the new node $X_{new}$ from $X_{nearest}$ towards $X_{rand}$ in a fixed step size. The algorithm detects collision based on the generated node $X_{new}$ and node $X_{nearest}$. Suppose the line segments between $X_{new}$ and $X_{nearest}$ are collision-free. In that case, the algorithm adds $X_{new}$ as a child node to the tree $T$. Otherwise, the algorithm discards $X_{new}$, the current iteration ends, and continues to the next iteration. The schematic diagram of the RRT algorithm is shown in Figure 3.

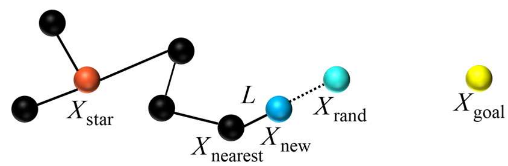



**Fig. 3.** Schematic diagram of RRT algorithm.

The RRT* algorithm improves on the basic RRT algorithm by introducing the cumulative cost attribute, which records the sum of the lengths of all edges on the path from the start point. It involves reselecting parent nodes and node reconnection to implement a new strategy. The schematic diagram of the RRT* algorithm is shown in Figure 4.

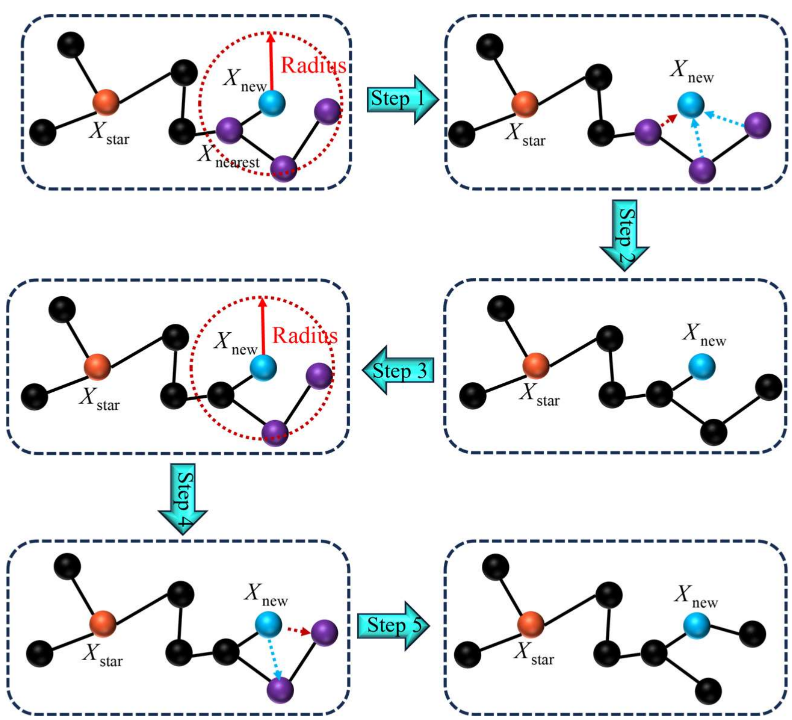



**Fig. 4.** Schematic diagram of RRT* algorithm.

Instead of using $X_{\text{nearest}}$ as the parent node in the RRT* algorithm, its selection process evaluates all nodes within a certain radius with $X_{\text{new}}$ as the centre point and selects the node with the lowest path cost as the optimal parent node of $X_{\text{new}}$. Once the optimal parent node is identified, the algorithm traverses the remaining tree nodes and calculates the path cost from each node to $X_{\text{new}}$ to $X_{\text{star}}$. The algorithm rebuilds the tree by selecting the path with the lowest cumulative cost.

### 3.2. *Bi-RRT**

When a single randomized tree is used to search the entire space, the search is inefficient. The RRT* algorithm is used to generate two random trees at $X_{\text{star}}$ and $X_{\text{goal}}$, denoted as $T_{\text{s}}$ and $T_g$. Two random trees grow alternately based on the idea of the RRT* algorithm, and the search is completed when the two trees are connected or are a certain distance away. Bi-RRT* can effectively improve the search efficiency, converge to the optimal solution faster, and generate the feasible path faster.

In Figure 5 below, the Bi-RRT* search process is visualized:

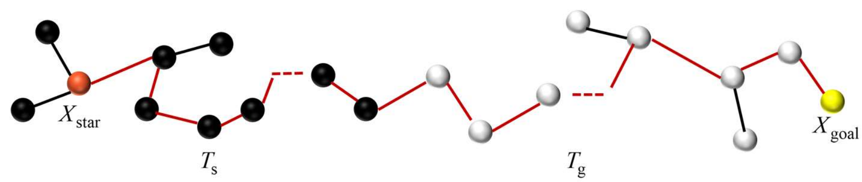



**Fig. 5.** Schematic diagram of Bi-RRT* algorithm.

The start point $X_{\text{star}}$ is the red ball on the left, and the target point $X_{\text{goal}}$ is the yellow ball on the right. The search is initiated from these two points at the same time, and each construct search trees $T_{\text{s}}$ and $T_g$, which grow alternately until a clear path from the start to the end point is established when the two trees are connected or when the two trees are a certain distance away from each other.

### 3.3. *Bi-Informed-RRT**

The Bi-Informed-RRT* algorithm is consistent with the Bi-RRT* algorithm in finding the initial path. However, after finding the initial path, the Bi-Informed-RRT* algorithm constructs an ellipsoid sampling domain based on the initial path length $C_{\text{best}}$ and the distance $C_{\text{min}}$ between $X_{\text{star}}$ and $X_{\text{goal}}$. This ellipsoid sampling domain is shown in Figure 6 below.

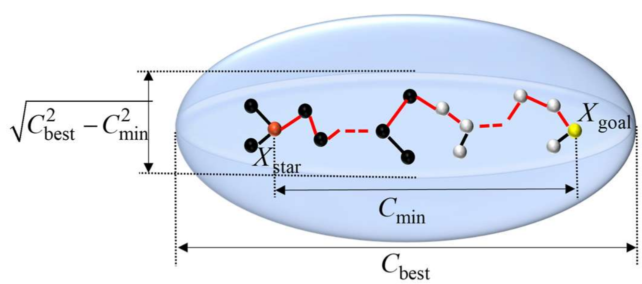



**Fig. 6.** Ellipsoid sampling domain.

$C_{\text{min}}$ is the distance between the two foci of the ellipsoid, $C_{\text{best}}$ is the length of the long axis of the ellipsoid, and $\sqrt{C_{\text{best}}^2 - C_{\text{min}}^2}$ is the length of the short and centre axes. The shape of this ellipsoid is uniform, and it is a rotating ellipsoid. The ellipsoid sampling domain eliminates sampling of invalid regions (regions outside the ellipsoid region). $C_{\text{min}}$ is the shortest path when there are no obstacles between the start and target points.

Bi-Informed-RRT* first samples in a spatial coordinate system and transforms the sampled points to ellipsoidal sampling regions by rotation and translation. Whenever a new node is generated, the search is accelerated by re-selecting the parent node and reconnecting the tree within the neighbourhood of the new node so that the path cost of certain nodes is lower, the path length is reduced, the global path length is reduced to $C_{best}^{'}$, and the sampling space is further reduced. The reconstructed ellipsoidal sampling domain is shown in Figure 7 below.

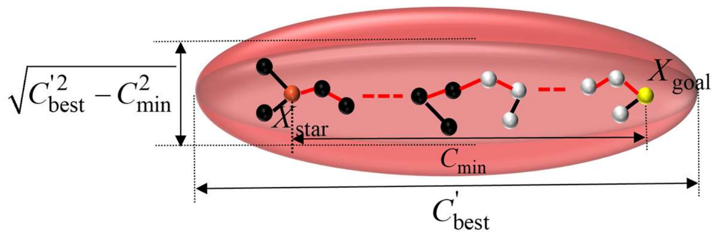



**Fig. 7.** Reconstructed ellipsoidal sampling domain.

## 4. Algorithmic improvements

### *4.1. Bi-Informed-RRT* algorithm for dual-path balancing operations strategy*

To realize the expansion of four random trees at the same time, this paper proposes the fusion of the dual-path balancing operations strategy with the Bi-Informed-RRT* algorithm (BD-Informed-RRT*). The whole path is divided into two path segments by dual-path balancing operation. The dual-path balancing operation means to add a middle root node $X_{middle}$ between $X_{star}$ and $X_{goal}$. the initial middle point is chosen to be the midpoint position of the line connecting the $X_{star}$ and the $X_{goal}$, with coordinates $P$. At the same time, it is necessary to consider whether the initial middle point is inside the obstacle.

1) The initial middle point is not inside the obstacle. This initial middle point is the final middle root node.

2) The initial middle point is inside the obstacle. The path is divided into two segments of equal length using vertical bisection. Make a plane $\gamma$ which crosses the initial middle point and is perpendicular to the line where $X_{star}$ and $X_{goal}$ are located. The plane $\gamma$ intersects the obstacle to obtain the two-dimensional obstacle boundary equation. Calculate the shortest distance from point $Q$ on the obstacle boundary to the initial middle point. Use $\boldsymbol{PQ}+\Delta\boldsymbol{PQ}$ as the final middle point. The middle point is determined as shown in Figure 8 below.

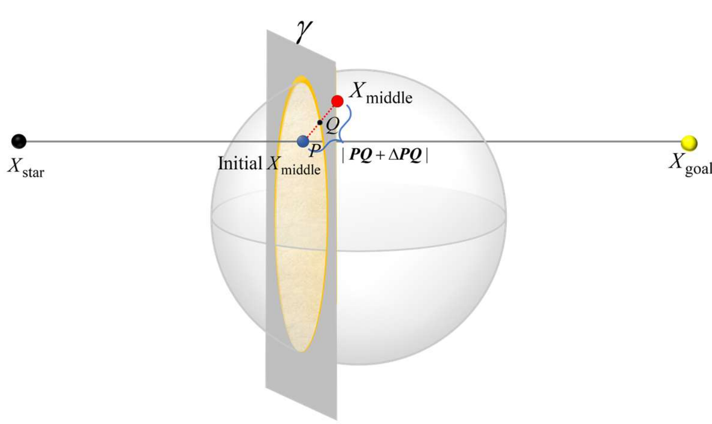



**Fig. 8.** Determination of the middle point.

The parallel expansion of four random trees ( $T_s$ , the random tree expanding from $X_{star}$ to $X_{middle}$ . $T_{m1}$ , the random tree expanding from $X_{middle}$ to $X_{star}$ . $T_{m2}$ , the random tree expanding from $X_{middle}$ to $X_{goal}$ . $T_g$ , the random tree expanding from $X_{goal}$ to $X_{middle}$ ) is realized by the dual-path balancing operation.

### 4.2. Bi-Informed-RRT* partition-biased sampling strategy

In applying the Bi-Informed-RRT* algorithm to generate initial paths in three-dimensional complex environments, its sampling strategy usually adopts simple random sampling in the global range, leading to greater randomness in the path search process. Path quality is positively correlated with the proximity of the connecting line between $X_{star}$ and $X_{goal}$ . In particular, if there are no obstacles on the path of the connecting line between $X_{star}$ and $X_{goal}$ , then the path formed along that connecting line will be the shortest. Based on this finding, this paper proposes a partition-biased sampling strategy fused with the BD-Informed-RRT* algorithm (BPD-Informed-RRT*), aiming to optimize the algorithm's performance and improve the efficiency and quality of path planning.

Firstly, the Bi-Informed-RRT* algorithm is used in the process of generating initial paths between $X_{star}$ and $X_{middle}$ and $X_{middle}$ and $X_{goal}$ . The partition-biased sampling strategy is added to each random tree to optimize the sampling process. The planning between $X_{star}$ and $X_{middle}$ is used here as an example to illustrate the methodology of the partition-biased sampling strategy. The partition-biased sampling strategy between $X_{middle}$ and $X_{goal}$ is the same. The global map is divided into two parts, edge region $A_1$ and centre region $A_2$ , and the partition-biased sampling strategy is shown in Figure 9.

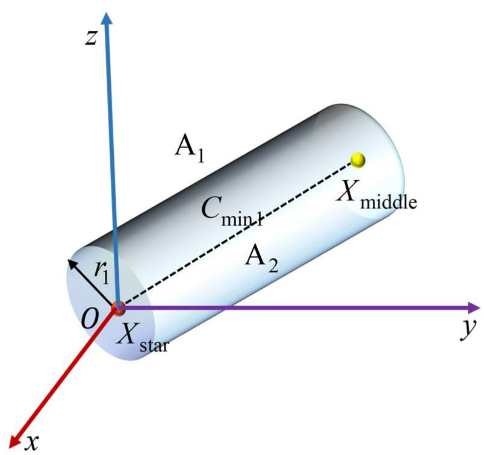



**Fig. 9.** Partition-biased sampling strategy.

In three-dimensional space, a cylinder is created. With the start point $X_{star}$ as the centre and $r_1$ as the radius, the lower bottom surface of the cylinder is established, and with the $X_{middle}$ as the centre and $r_1$ as the radius, the upper bottom surface of the cylinder is established, and the height of the cylinder is the distance $C_{min1}$ between the $X_{star}$ and $X_{middle}$ .The centre region $A_2$ is the inner region of the cylinder, and the region in the three-dimensional space excluding the centre region is the edge region $A_1$ .

The partition-biased sampling strategy is:

$$X_{rand}=\begin{cases} A_1, rand(1)>P_1 \begin{cases} X_{star}/X_{middle}, rand(1)>P_2 \\ X_{free-edge} \quad , rand(1)\le P_2 \end{cases} \\ A_2, rand(1)\le P_1 \begin{cases} X_{star}/X_{middle}, rand(1)>P_2 \\ X_{free-centre} \quad , rand(1)\le P_2 \end{cases} \end{cases} \tag{7}$$

In equation (7): $X_{\text{rand}}$ is the sampling point. $X_{\text{star}}$ is the start point. $X_{\text{middle}}$ is the middle point. $X_{\text{free-edge}}$ is the random point within the edge free region. $X_{\text{free-centre}}$ is the random point within the centre free region. P1 and P2 are the bias probabilities.

When the first random number $\text{rand}(1) > P_1$ and the second random number $\text{rand}(1) > P_2$, the sampling point is taken as the $X_{\text{star}} / X_{\text{middle}}$ of the two-way search random tree. When the first random number $\text{rand}(1) > P_1$ and the second random tree $\text{rand}(1) \le P_2$, take the sampling point as a random point inside the edge free region $A_1$.

When the first random number $\text{rand}(1) \le P_1$ and the second random number $\text{rand}(1) > P_2$, the sampling point is taken as the $X_{\text{star}} / X_{\text{middle}}$ of the two-way search random tree. When the first random number $\text{rand}(1) \le P_1$ and the second random tree $\text{rand}(1) \le P_2$, take the sampling point as a random point inside the centre free region $A_2$.

The partition-biased strategy optimizes the path search process by increasing the sampling density near the connecting line between $X_{\text{star}}$ and $X_{\text{middle}}$. Moreover, it enhances the target orientation and improves the efficiency of the Bi-Informed-RRT* algorithm in initial path generation.

Secondly, The Bi-Informed-RRT* algorithm for path optimization also introduces the partition-biased sampling strategy. The sampling ellipsoid is divided into edge region A1 and centre region A2, and a cylinder is introduced in the sampling ellipsoid. The centre region A2 is the inner region of the cylinder, and the region in the sampling ellipsoid space excluding the centre region is the edge region A1. The partition-biased sampling strategy is shown in Figure 10.

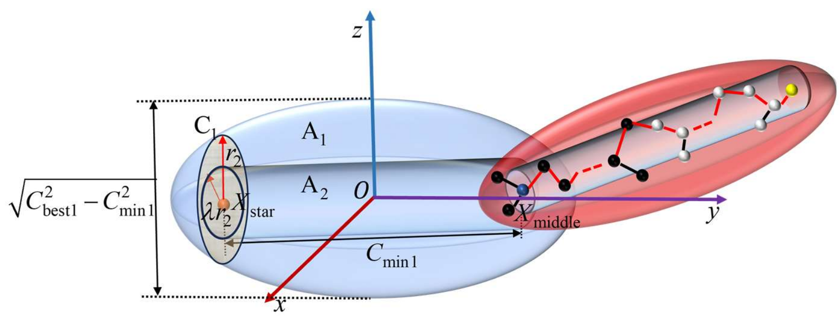



**Fig. 10.** Partition-biased sampling strategy.

The sampling ellipsoid space will continue to decrease as the path quality continues to be optimized. The design of the cylinder needs to consider the variation of the ellipsoid space, so this paper needs to define the radius and height parameters of the upper bottom surface and lower bottom surface of the cylinder. Here, the path optimization of the Bi-Informed-RRT* algorithm between $X_{\text{star}}$ and $X_{\text{middle}}$ is an example. The cylinder parameters are defined as follows:

In a three-dimensional coordinate system, assume that the centre of the sampling ellipsoid is located at (0, 0, 0), the length of the long axis of the ellipsoid is $C_{\text{best1}}$, and the lengths of the short and intermediate axes are $\sqrt{C_{\text{best1}}^2 - C_{\text{min1}}^2}$. The equations of this ellipsoid can be expressed as:

$$\frac{x^2}{\sqrt{C_{\text{best1}}^2 - C_{\text{min1}}^2}} + \frac{y^2}{C_{\text{best1}}^2} + \frac{z^2}{\sqrt{C_{\text{best1}}^2 - C_{\text{min1}}^2}} = 1 \tag{8}$$

A plane perpendicular to the long axis through the starting point $X_{\text{star}}$ intersects the ellipsoid to form a circle $C_1$, and the equation of this plane is $y = C_{\text{min1}}/2$. The equation of the circle $C_1$ formed by the intersection of the ellipsoid and the plane by bringing $y = C_{\text{min1}}/2$ into the ellipsoid equation solvable is:

$$x^2 + z^2 = C_{\text{best1}}^2 - \frac{5C_{\min 1}^2}{4} + \frac{C_{\min 1}^4}{4C_{\text{best1}}^2} \tag{9}$$

Let: $C_{\text{best1}}^2 - \frac{5C_{\min 1}^2}{4} + \frac{C_{\min 1}^4}{4C_{\text{best1}}^2} = r_2^2$

Then: $x^2 + z^2 = r_2^2$

The radius of $C_1$ can be obtained as $r_2$ .

Create a cylinder in the sampled ellipsoid space. With the $X_{\text{star}}$ as the centre of the circle, $\lambda r_2$ as the radius ( $\lambda \in (0,1)$ , is the radius coefficient), the lower bottom surface of the cylinder is created. The upper bottom surface of the cylinder is created with the $X_{\text{middle}}$ as the centre and $\lambda r_2$ as the radius. The height of the cylinder is the distance $C_{\min 1}$ between the $X_{\text{star}}$ and the $X_{\text{middle}}$.

The partition-biased sampling strategy is the same as equation (7) above.

### *4.3. BPD-APF-Informed-RRT* algorithm*

The BD-Informed-RRT* algorithm processed by the partition-biased sampling strategy significantly improves the performance of the optimization algorithm. It improves the efficiency of path planning compared with the traditional algorithm. However, the generation of sampling points still has randomness. In order to further accelerate the search efficiency of random trees, this paper proposes the fusion algorithm of the artificial potential field method and BPD-Informed-RRT* (BPD-APF-Informed-RRT*) to adjust the generation direction of new nodes.

The core idea of APF is to abstract the environment around the UAV as a virtual potential field. The principle of the artificial potential field method is shown in Figure 11.

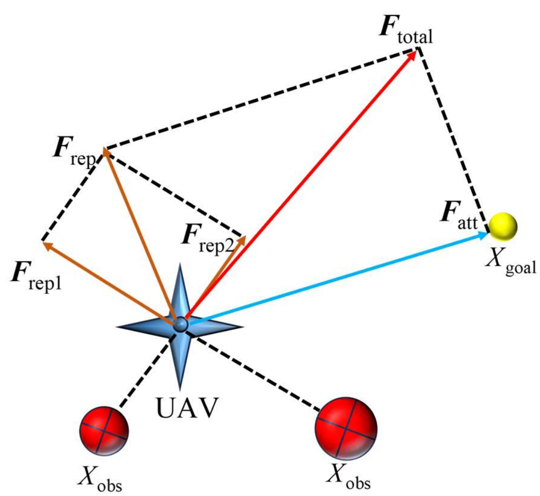



**Fig. 11.** Schematic diagram of artificial potential field method.

In this potential field, the target point exerts a continuous attractive force on the UAV, while obstacles exert a repulsive force within their sphere of influence.

4.3.1. APF and Bi-Informed-RRT* algorithm fusion. The improvement of APF fused with the Bi-Informed-RRT* algorithm in this paper has the advantages of simple structure and fast information processing. The details are as follows: Firstly, sampling point $X_{\text{rand}}$ is generated in the sampling space by partition-biased sampling strategy, and the $X_{\text{rand}}$ coordinates information is determined. Then, under the premise of $X_{\text{rand}} \neq X_{\text{star}} / X_{\text{goal}}$, the target point $X_{\text{goal}}$ is set as the attraction point, which generates continuous attraction to $X_{\text{rand}}$ . The obstacle produces a repulsive force on $X_{\text{rand}}$ . A guided point $X_{\text{guided}}$ is generated with the length of $\eta|\boldsymbol{F}_{\text{total}}|$ in the direction where $X_{\text{rand}}$ is subjected to the combined force. This guided point $X_{\text{guided}}$ replaces the $X_{\text{rand}}$ to extend a new node. The principle of $X_{\text{guided}}$ generation is shown in Figure 12 below.

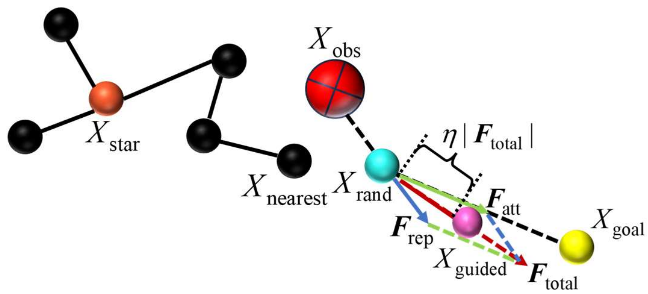



**Fig. 12.** Principle of $X_{\text{guided}}$ generation.

The coordinates of the final guided point $X_{\text{guided}}$ can be calculated using the following equation:

$$X_{\text{guided}} = X_{\text{rand}} + \eta \boldsymbol{F}_{\text{total}} \tag{10}$$

In equation (10): $\eta$is the guidance coefficient, which controls where the guidance points are generated.

Specifically, the APF consists of two main components: the gravitational field function $\boldsymbol{U}_{\text{att}}(q_{\text{r}})$ and the repulsive field function $\boldsymbol{U}_{\text{rep}}(q_{\text{r}})$, whose expressions are shown in equation (11)and equation(12). The total potential field function $\boldsymbol{U}_{\text{total}}$ is shown in equation(13).

$$\boldsymbol{U}_{\text{att}}(q_{\text{r}}) = \frac{1}{2} K_{\text{att}} \rho(q_{\text{r}}, q_{\text{g}}) \tag{11}$$

$$\boldsymbol{U}_{\text{rep}}(q_{\text{r}}) = \begin{cases} \frac{1}{2} K_{\text{rep}} \left( \frac{1}{\rho(q_{\text{r}}, q_{\text{obs}})} - \frac{1}{\rho_0} \right)^2 & , \rho(q_{\text{r}}, q_{\text{obs}}) \le \rho_0 \\ 0 & , \rho(q_{\text{r}}, q_{\text{obs}}) > \rho_0 \end{cases} \tag{12}$$

$$\boldsymbol{U}_{\text{total}} = \sum \boldsymbol{U}_{\text{att}} + \sum \boldsymbol{U}_{\text{rep}} \tag{13}$$

In equation (11), (12) and (13): $K_{\text{att}}$ and $K_{\text{rep}}$ are the gravitational gain coefficient and repulsive gain coefficient. $q_{\text{r}}$ is the position of the sampling point $X_{\text{rand}}$. $q_{\text{obs}}$ is the position of the obstacle. $q_{\text{g}}$ is the position of the target point. $\rho(q_{\text{r}}, q_{\text{g}})$ is the Euclidean distance between the $X_{\text{rand}}$ and the target point $X_{\text{goal}}$, $\rho(q_{\text{r}}, q_{\text{obs}})$ is the Euclidean distance between the $X_{\text{rand}}$ and the obstacle. $\rho_0$ is the maximum influence distance of the obstacle to the $X_{\text{rand}}$.

The negative gradient of the gravitational and repulsive field functions can express the magnitude of the gravitational and repulsive forces. The vector sum of the gravitational and repulsive forces then determines the magnitude of the combined force. The magnitudes of the gravitational force, the repulsive force, and the combined force are shown in equations(14), (15) and (16).

$$\boldsymbol{F}_{\text{att}}(q_{\text{r}}) = -\nabla \boldsymbol{U}_{\text{att}}(q_{\text{r}}) = K_{\text{att}} \rho(q_{\text{r}}, q_{\text{g}}) \tag{14}$$

$$\boldsymbol{F}_{\text{rep}}(q_{\text{r}}) = -\nabla \boldsymbol{U}_{\text{rep}}(q_{\text{r}}) = \begin{cases} K_{\text{rep}} \left( \frac{1}{\rho(q_{\text{r}}, q_{\text{obs}})} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q_{\text{r}}, q_{\text{obs}})} & , \rho(q_{\text{r}}, q_{\text{obs}}) \le \rho_0 \\ 0 & , \rho(q_{\text{r}}, q_{\text{obs}}) > \rho_0 \end{cases} \tag{15}$$

$$F_{total} = \sum F_{att} + \sum F_{rep} \tag{16}$$

In order to solve the problem of target unreachability in APF. This paper uses the improved repulsive field function in literature. The improved repulsive field function $U_{rep}(q_r)$ and the magnitude of the repulsive force $F_{rep}(q_r)$ are shown in equation (17) and (18).

$$U_{rep}(q_r) = \begin{cases} \frac{1}{2} K_{rep} \left( \frac{1}{\rho(q_r, q_{obs})} - \frac{1}{\rho_0} \right)^2 \rho^t(q_r, q_g), & \rho(q_r, q_{obs}) \le \rho_0 \\ 0 & , \rho(q_r, q_{obs}) > \rho_0 \end{cases} \tag{17}$$

$$F_{rep}(q_r) = -\nabla U_{rep}(q_r) = \begin{cases} F_{rep1}(q_r) + F_{rep2}(q_r), & \rho(q_r, q_{obs}) \le \rho_0 \\ 0 & , \rho(q_r, q_{obs}) > \rho_0 \end{cases} \tag{18}$$

In equation (18):

$$F_{rep1}(q_r) = K_{rep} \left( \frac{1}{\rho(q_r, q_{obs})} - \frac{1}{\rho_0} \right) \frac{\rho^t(q_r, q_g)}{\rho^2(q_r, q_{obs})} \tag{19}$$

$$F_{rep2}(q_r) = -\frac{t}{2} K_{rep} \left( \frac{1}{\rho(q_r, q_{obs})} - \frac{1}{\rho_0} \right)^2 \rho^{t-1}(q_r, q_g) \tag{20}$$

Where: $t$ is the adjustment factor and is a positive integer.

The new repulsive force on the $X_{rand}$ in the obstacle-influenced region is synthesized by the repulsive force in two directions, where $F_{rep1}$ is the repulsive force of the obstacle pointing to the $X_{rand}$, and $F_{rep2}$ is the repulsive force of the $X_{rand}$ pointing to the target point $X_{goal}$.

4.3.2. Repulsive potential field calculation based on different obstacle models. In the literature, the obstacles in path planning are regular spherical obstacles, only modeled as spherical obstacles. However, in the actual flight of a UAV, the obstacles encountered are mostly irregular three-dimensional objects, such as buildings and trees. In this paper, spherical and cylindrical obstacles are modeled to satisfy the path planning of UAVs in complex environments. A repulsive potential field calculation method is designed based on the nearest point of different obstacle models.

The coordinate vector of the current position of the sampling point $X_{rand}$ is $(x_{rand}, y_{rand}, z_{rand})$, and the nearest point algorithm based on different obstacles is as follows:

1) Sphere obstacle. The sphere surface model can be expressed as:

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = r_c^2 \tag{21}$$

In equation (21): $O_c(x_c, y_c, z_c)$ is the centre point coordinates of the sphere obstacle. $q_{obs}(x, y, z)$ is the edge point coordinates of the sphere obstacle. $r_c$ is the radius of the sphere obstacle.

The coordinates of the position of the nearest obstacle point on the surface of a spherical obstacle are calculated as shown in Figure 13.

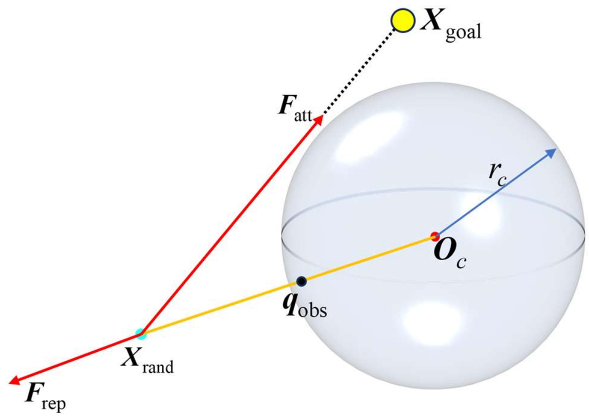



**Fig. 13.** Schematic diagram of the surface obstruction points of a sphere obstacle.

The formula can express the coordinates of the position of the nearest obstacle point:

$$q_{obs} = r_c \frac{X_{rand} - O_c}{\rho(X_{rand}, O_c)} + O_c \tag{22}$$

2) Cylindrical obstacles. The side surfaces model of the cylinder can be expressed as:

$$(x-x_a)^2 + (y-y_a)^2 = r_a^2, z-z_a < h_a \tag{23}$$

In equation (23): The coordinates of the centres of the upper and lower surfaces of the cylinder are $O_{ah}(x_a, y_a, z_a + h_a)$ and $O_a(x_a, y_a, z_a)$, respectively. $q_{obs}(x, y, z)$ is the coordinate of a point on the side edge of the cylinder. $h_a$ is the height of the cylinder. $r_a$ is the radius of the base.

The coordinates of the position of the nearest obstacle point on the surface of a cylindrical obstacle are calculated as shown in Figure 14.

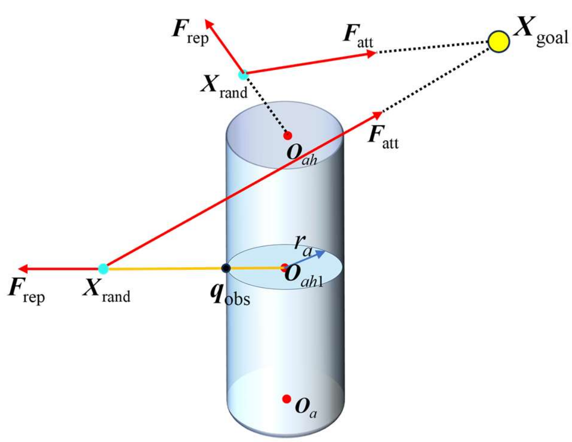



**Fig. 14.** Schematic diagram of the surface obstruction points of a cylindrical obstacle.

When $0 \le z_{rand} \le h_a$, the problem of finding the nearest point of $X_{rand}$ to the edge of the cylinder can be downscaled by finding the nearest point $q_{obs}$ on the circle in a planar coordinate system at the same height as $X_{rand}$.

When $z_{rand} > h_a$, take $q_{obs} = O_{ah}$.

The coordinates of the position of the nearest obstacle point can be expressed by the formula:

$$\begin{cases} \boldsymbol{q}_{\text{obs}} = r_a \dfrac{\boldsymbol{X}_{\text{rand}} - \boldsymbol{O}_{ah1}}{\rho(\boldsymbol{X}_{\text{rand}}, \boldsymbol{O}_{ah1})} + \boldsymbol{O}_{ah1}, 0 \le z_{\text{rand}} \le h_a \\ \boldsymbol{q}_{\text{obs}} = \boldsymbol{O}_{ah} \qquad , z_{\text{rand}} > h_a \end{cases} \tag{24}$$

In equation (24): $\boldsymbol{O}_{ah1}$ is the coordinates of the location point on the centreline of the cylinder that is equal in height to $\boldsymbol{X}_{\text{rand}}$. $\boldsymbol{O}_{ah}$ is the coordinate of the centre of the circle on the upper surface of the cylindrical obstacle.

### 4.4. BPD-APF-Informed-FARRT* algorithm

In the spatial exploration process of the BPD-APF-Informed-RRT* algorithm, a fixed step size can adversely affect the algorithm's search efficiency. When facing an environment with dense obstacles, a fixed step size may cause the search process to be challenging to find a suitable expansion step, which in turn causes delays or interruptions in path construction. On the other hand, in scenarios with fewer obstacles or obstacles farther away, a constant step size may lead to unnecessarily prolonging the search time and increasing the complexity of the path.

In order to balance the problem of too large or too small step size in the path search process, this paper introduces fuzzy control to adjust the step size adaptively. The step size value can be dynamically adjusted according to the density of obstacles around the node $X_{\text{nearest}}$ nearest to the guided point $X_{\text{guided}}$ $(ObsDensity)$ and the distance from the node $X_{\text{nearest}}$ to the target point $X_{\text{goal}}$ $(PointDist)$.

The fuzzy control method, based on fuzzy set theory, fuzzy linguistic variables, and fuzzy logic inference, realizes dynamic mapping between input and output by transforming engineering experience and rules into fuzzy rules. The architecture of the fuzzy controller is divided into four core components: fuzzification, knowledge base, fuzzy inference, and defuzzification. It is shown in Figure 15.

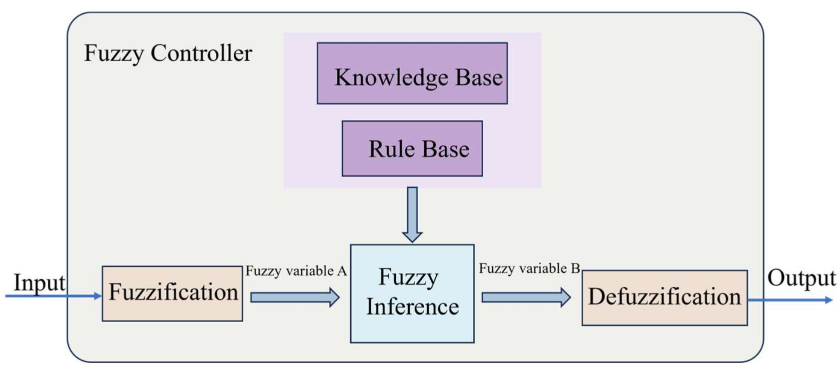



**Fig. 15.** Fuzzy controller structure diagram

4.4.1. Determination of fuzzy controller variables. First, $ObsDensity$ is used as an essential input variable to evaluate the complexity of the environment. In regions with dense obstacles, the fuzzy controller decreases the step size to achieve an accurate search in a limited space. This adjustment facilitates detailed path exploration in narrow or crowded environments. On the contrary, in regions with fewer obstacles, the fuzzy controller increases the step length to improve the speed and efficiency of the search. Second, $PointDist$ is another control parameter, and the step size is adjusted according to the distance size. Finally, the adaptive step size coefficient $\mu$ is used as the output of the fuzzy controller. $\mu X_{\text{nearest}} X_{\text{guided}}$ determines the position of the expanded new node $X_{\text{new}}$.

The schematic diagram of adaptive step size adjustment is shown in Figure 16 below.

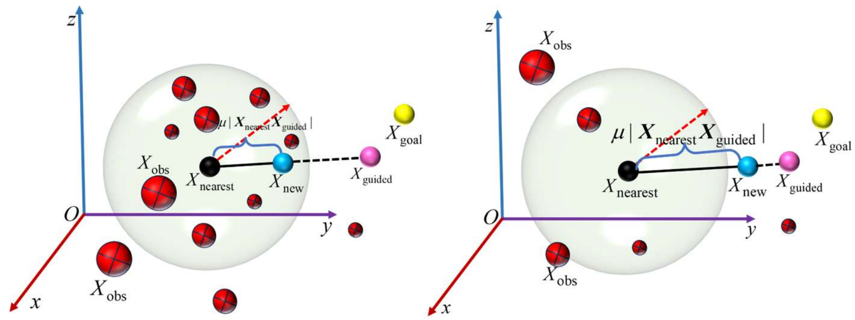



**Fig. 16.** Adaptive step size adjustment.

From Figure 16, it can be seen that when there are fewer obstacles and far away from the target point, the output $\mu$ value of the fuzzy controller is larger, and the expansion step of the new node is larger. So, only a few times of sampling are needed to reach the target point.

4.4.2. Determination of the membership function. The membership function is a critical component in fuzzy logic. In designing the membership function of a fuzzy controller, the key lies in determining the number of fuzzy sets and the shape of its membership function. This paper chooses to use the gaussmf and trimf membership functions, which can simplify the design process and improve the system performance due to their high computational efficiency and fast response time.

After normalizing $ObsDensity$ and $PointDist$, and $\mu$, their ranges of values are [0,5] and [0,10], and [0,1], respectively.

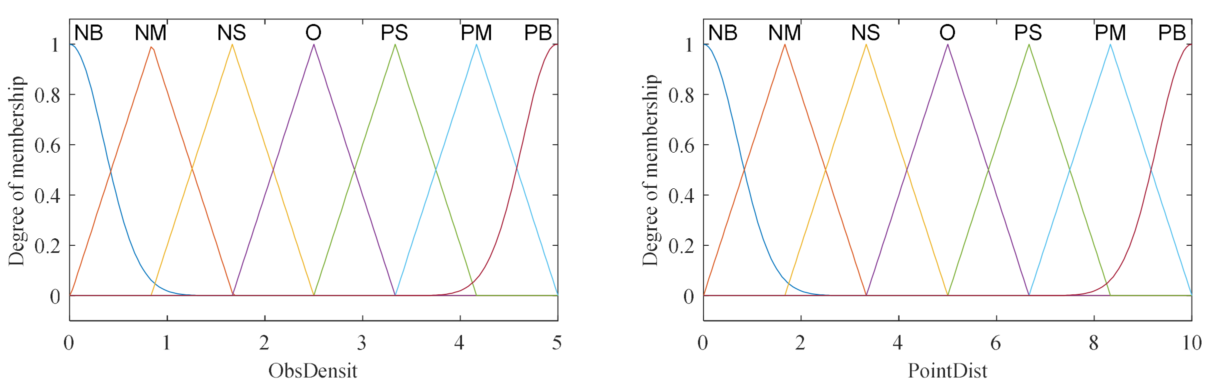
The input/output linguistic variables are defined as:

$$\begin{cases} ObsDensity \equiv \{NB, NM, NS, O, PS, PM, PB\} \\ PointDist \equiv \{NB, NM, NS, O, PS, PM, PB\} \\ \mu \equiv \{NB, NM, NS, O, PS, PM, PB\} \end{cases} \tag{25}$$

Note:

$$\begin{array}{lll} \text{Negative} \equiv \text{N} & \text{Positive} \equiv \text{P} & \text{Zero} \equiv \text{O} \\ \text{Big} \equiv \text{B} & \text{Medium} \equiv \text{M} & \text{Small} \equiv \text{S} \end{array}$$

Figure 17 shows the membership function of the input and output variables.



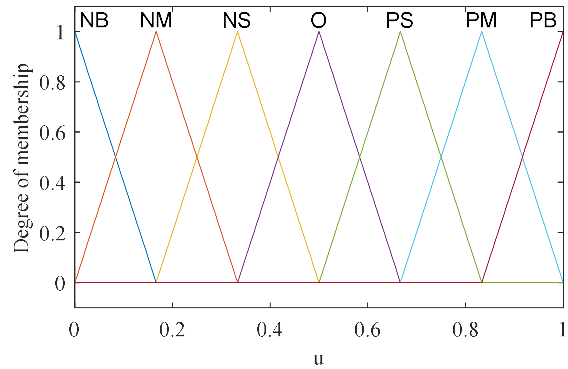



**Fig. 17.** Membership function of the input and output variables.

4.4.3. Establishing fuzzy control rules and fuzzy inference. Fuzzy inference using Mamdani method, if *ObsDensity* is A and *PointDist* is B, then $\mu$ is C. The step size is fine-tuned according to the density of obstacles around $X_{\text{nearest}}$ and the distance from $X_{\text{nearest}}$ to the target point to reduce unnecessary search time and path complexity. Moreover, fuzzy rules are formulated as shown in Table 1 below.

**Table 1.** Fuzzy rule table.

| *Obs - Density* | *PointDist* | | | | | | |
|---|---|---|---|---|---|---|---|
| | NB | NM | NS | O | PS | PM | PB |
| NB | O | PS | PS | PM | PM | PB | PB |
| NM | NS | O | PS | PS | PM | PM | PB |
| NS | NS | NS | O | PS | PS | PM | PM |
| O | NM | NS | NS | O | PS | PS | PM |
| PS | NM | NM | NS | NS | O | PS | PS |
| PM | NB | NM | NM | NS | NS | O | PS |
| PB | NB | NB | NM | NM | NS | NS | O |

In defuzzification, the centre of gravity method obtains the weighted average of the output fuzzy sets. The fuzzy inference surface is obtained according to the above fuzzy rules, as shown in Figure 18.

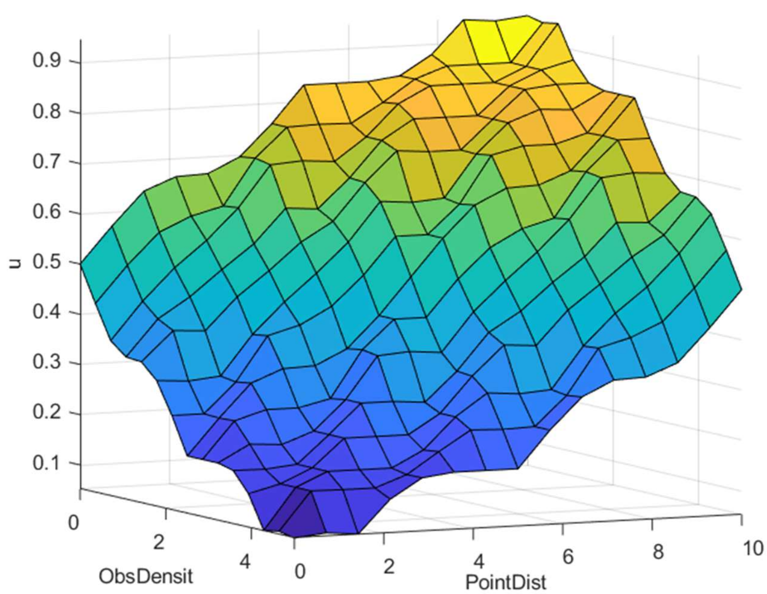



**Fig. 18.** Fuzzy inference surface.

### 4.5. Path pruning and optimization

With the partition-biased sampling and artificial potential field, the search tree reduces the search for redundant space, and the redundant nodes are reduced accordingly. However, there are still more redundant nodes in the path. Firstly, the triangular inequality principle removes the redundant nodes and generates the shortest path. Secondly, considering the kinematic characteristics of the UAV, it is necessary to constrain the smoothness of the path after redundancy. B-spline curve smoothing constraint is a common method that can generate smooth paths with continuous curvature by using the path points as the control points of the base function of the B-spline curve.

Then, the mathematical expression for the k-order B-spline curve containing n+1 control points $P_i(i=0,1,2,\ldots,n)$ and node vectors $\boldsymbol{N}=[t_0,t_1,\ldots,t_{n+k+1}]$ is:

$$P(t)=\sum_{i=0}^{n} P_i B_{i,k}(t) \tag{26}$$

In equation (26): $P(t)$ is the B-spline curve function, and $B_{i,k}(t)$ denotes the $i$th-order B-spline curve basis function. The B-spline curve basis function can be derived from the De Boor-Cox recursive equation as follows:

$$\begin{cases} B_{i,k}(t)=\begin{cases} 1, t_i \le t \le t_{i+1} \\ 0, \text{other} \end{cases}, k=0 \\ B_{i,k}(t)=\dfrac{t-t_i}{t_{i+k-1}-t_i} B_{i,k-1}(t)+\dfrac{t_{i+k}-t}{t_{i+k}-t_{i+1}} B_{i+1,k-1}(t), k \ne 0 \end{cases} \tag{27}$$

In order to satisfy the constraint that the curve passes through the start endpoint and the target endpoint, the node vectors of the two end nodes have a repetition degree k+1 as follows:

$$\begin{cases} t_0=t_1=t_2=\cdots=t_k \\ t_{n+1}=t_{n+2}=\cdots=t_{n+k+1} \end{cases} \tag{28}$$

A B-spline curve can be constructed given parameters such as control points, node vectors, and number of curves.

Figure 19 shows the smoothing of the path by a 3rd-order B-spline curve with selected control points [0 0 0; 1 2 2; 2 3 5; 3 5 6; 5 8 8; 8 8 8]. In order to satisfy the constraint that the curve passes through the start endpoint and the target endpoint, the node vectors of the two end nodes have a repetition degree 4. The node vectors of the two end nodes are selected as [0,0,0,0,0,0.3,0.7,1,1,1,1].

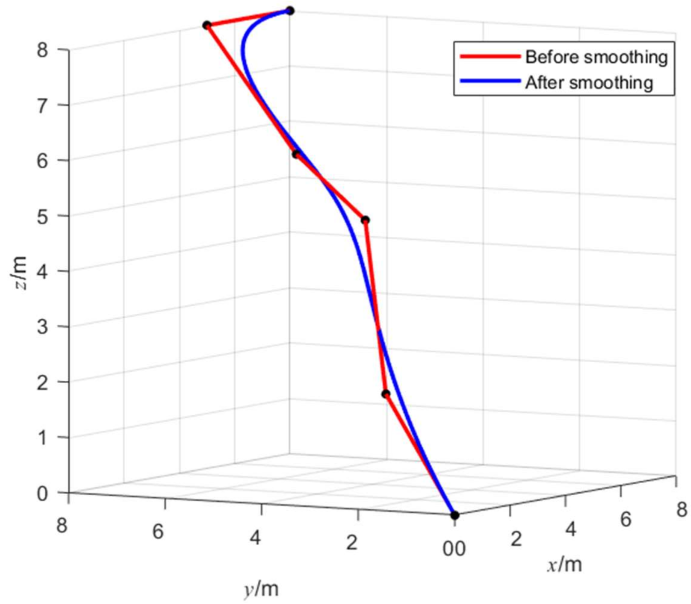
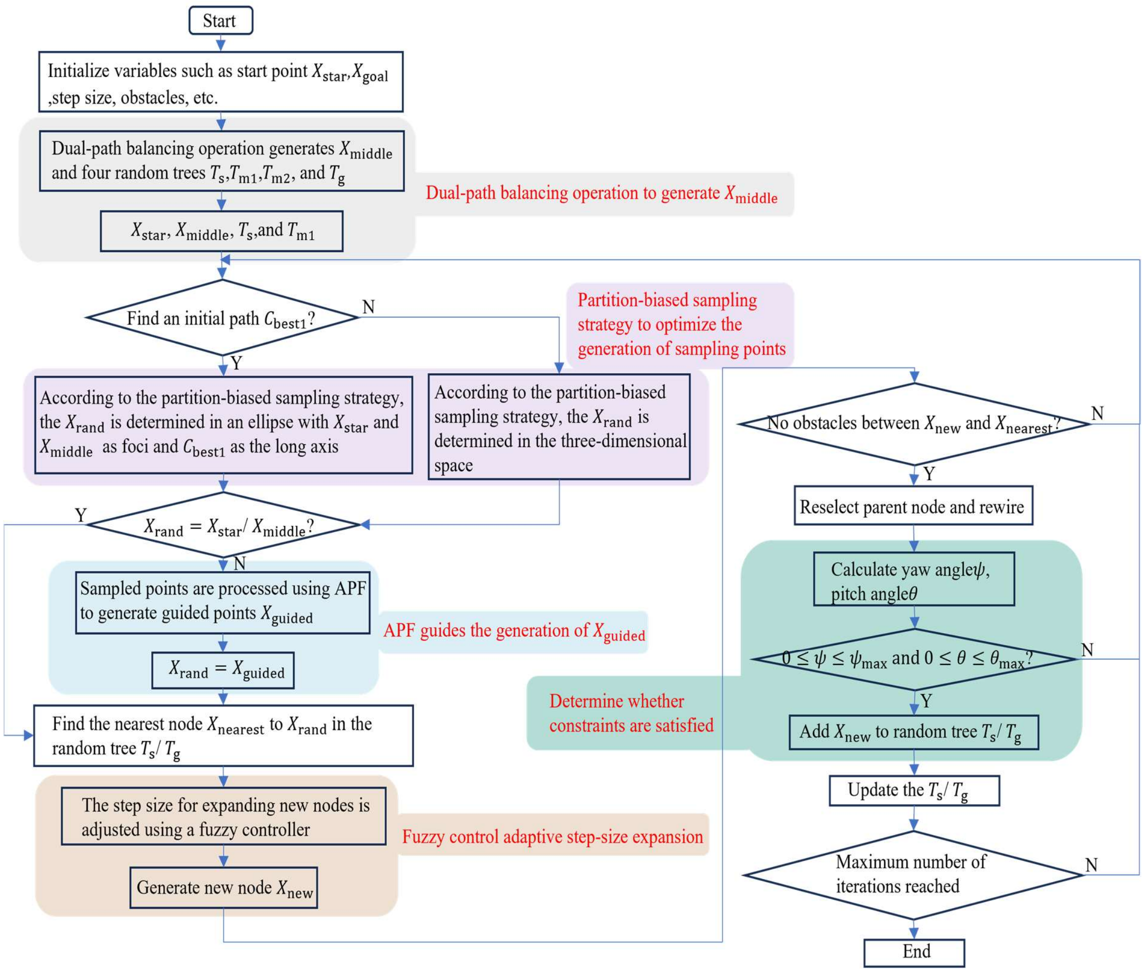



**Fig. 19.** Smoothing of the path.

### *4.6. Improved algorithm flowchart*

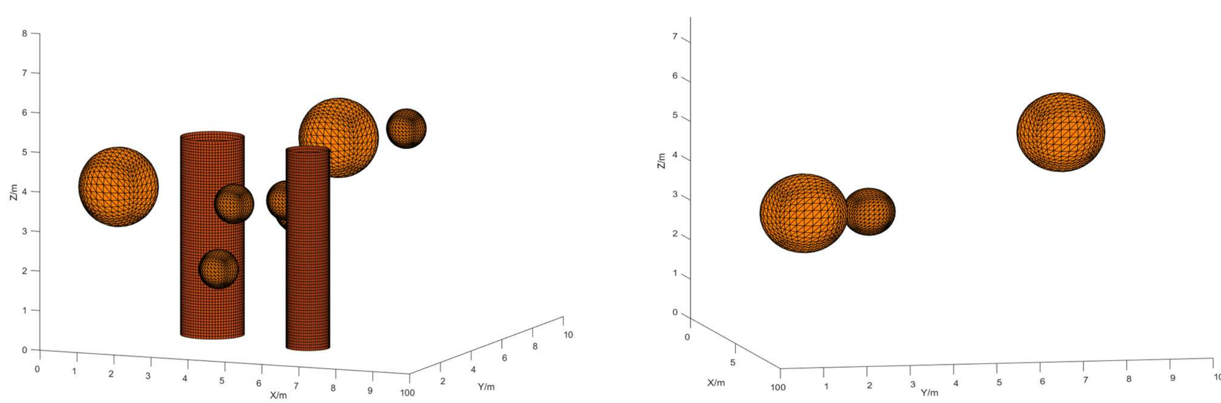
The planning between $X_{\text{star}}$ and $X_{\text{middle}}$ is used here as an example to illustrate the flow principle of the algorithm. The flow between the $X_{\text{middle}}$ and $X_{\text{goal}}$ is the same.



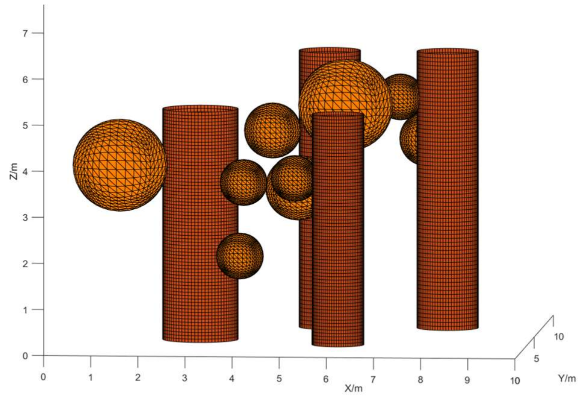

## 5. Simulation verification and analysis

This paper used a computing platform with an AMD R7-5800H processor and a Windows 11 operating system. It also conducted simulation experiments using MATLAB R2020a software.

The comparative analysis of the BPD-APF-Informed-FARRT* with Bi-RRT*, Informed-RRT*, and Bi-Informed-RRT* algorithms (we abbreviate them as 'BAIFR', 'BR', 'IR' and 'BIR', respectively, in the subsequent figures and tables.), the improved BPD-APF-Informed-FARRT* algorithm is proved to have the advantages of fast convergence and short paths, and the superior performance and feasibility of the proposed algorithm are verified. The simulation environment is a three-dimensional space of 10 m × 10 m × 10 m. The UAV starts from point (0,0,2) and ends at point (10,10,5). The algorithm in this paper is simulated in three environments, as shown in Figure 20.



(a) Environment A (b) Environment B




(c) Environment C

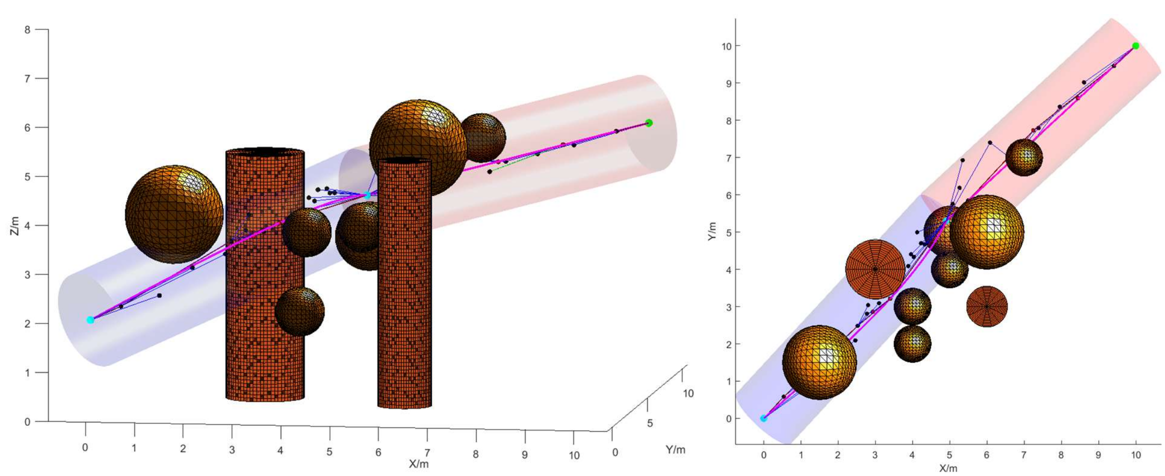
**Fig. 20.** Simulation environment.

The approximate parameter settings for the experiments in this paper are as follows: yaw and pitch angle constraints range from: $0 \le \psi \le \pi/2$ , $0 \le \theta \le \pi/2$ . $r_1 = 2\text{m}$ in the partition-biased sampling strategy. bias probability $P_1 = 0.7$ , $P_2 = 0.3$ . the basic step size of the comparison algorithm is 1 m. The gravitational gain coefficient $K_{att}$ is 0.2, the repulsive gain coefficient $K_{rep}$ is 0.06, and the guidance coefficient $\eta$ is 0.7.

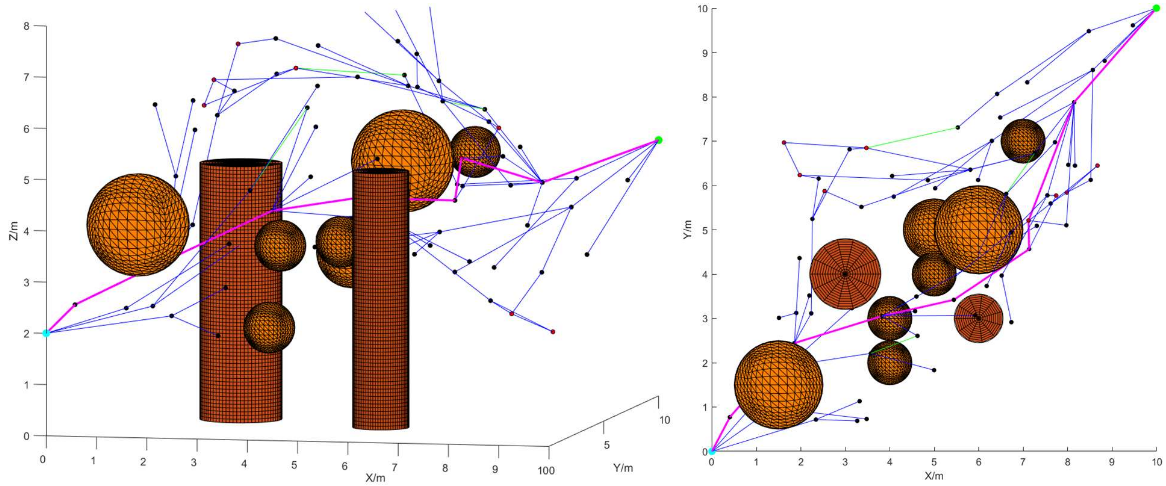
In the figure below, brown spheres and cylinders are obstacles, translucent blue and red cylinders are cylinder sampling domains for the partition-biased sampling strategy, translucent blue and red ellipsoids are ellipsoid sampling domains, cyan and green dots represent start and target points, respectively, black dots are nodes in the tree, red dots are rerouted parent nodes, blue line is a branch of the randomized tree, green line is a rerouted branch, dark red line is the final path.

## *5.1. Environment A*

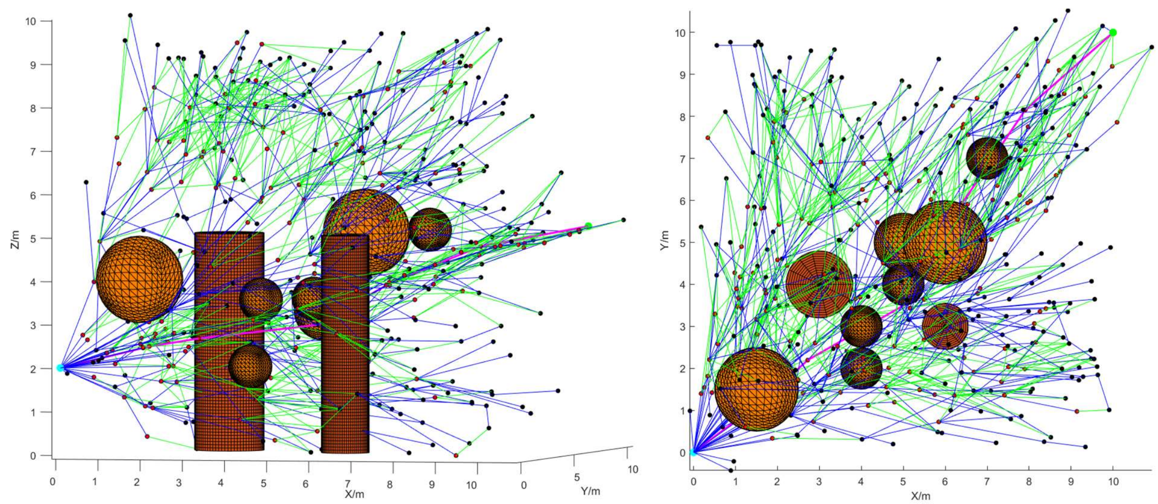
The results of the BPD-APF-Informed-FARRT* algorithm with the Bi-RRT*, Informed-RRT*, and Bi-Informed-RRT* algorithms for generating the initial paths are compared in environment A, with each algorithm being run 30 times in the environment A. There are seven brown sphere obstacles with different radius and two cylindrical obstacles in environment A. The initial path planning results of the four algorithms are shown in Figure 21.



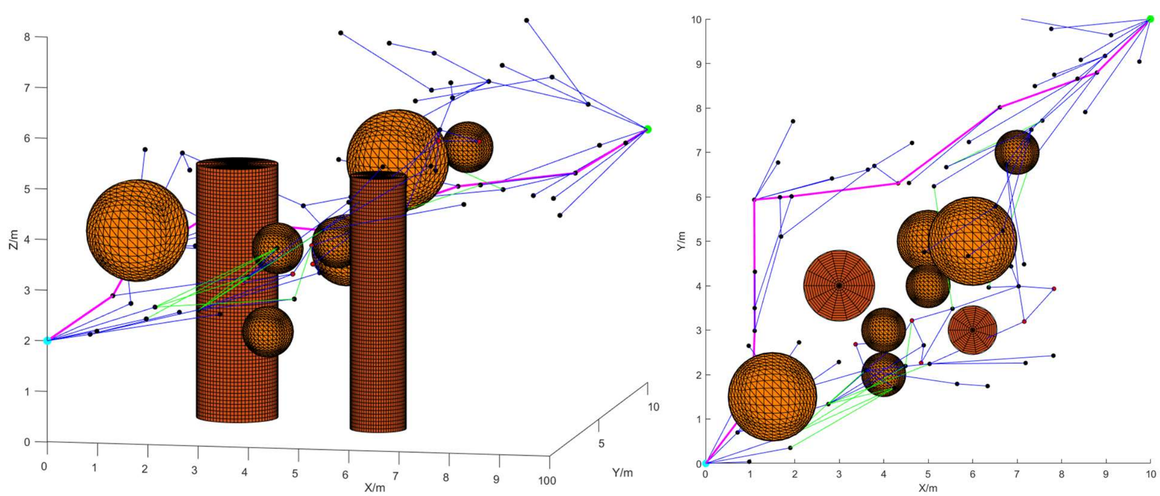

(a) Front and top views of the initial path of the BAIFR algorithm




(b) Front and top views of the initial path of the BR algorithm




(c) Front and top views of the initial path of the IR algorithm



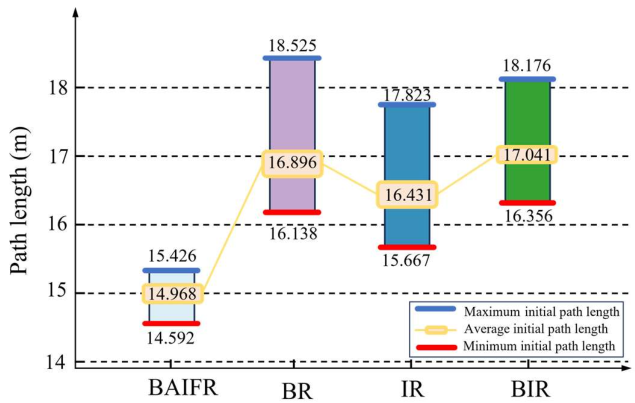

(d) Front and top views of the initial path of the BIR algorithm

**Fig. 21.** Initial path planning effect in environment A.

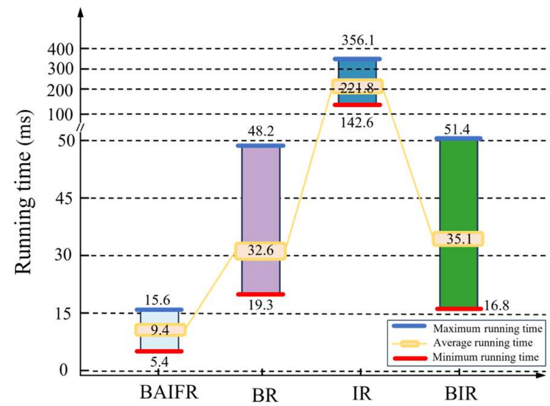
Comparative experimental results are shown in Figure 22 and Table 2.

**Table 2.** Simulation date in environment A.

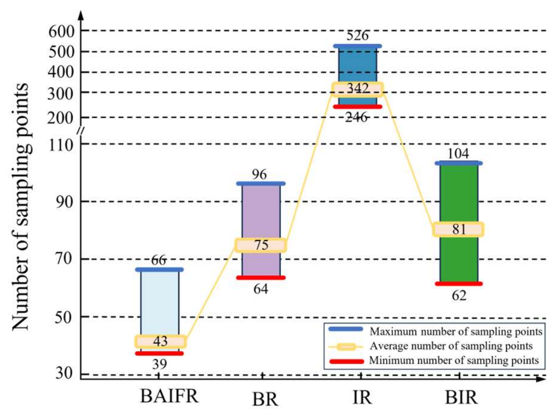
| Algorithm | Path length (m) | | | Running time (ms) | | | Number of sampling points | | |
|---|---|---|---|---|---|---|---|---|---|
| | Max | Average | Min | Max | Average | Min | Max | Average | Min |
| BAIFR | 15.426 | 14.968 | 14.592 | 15.6 | 9.4 | 5.4 | 66 | 43 | 39 |
| BR | 18.525 | 16.896 | 16.138 | 48.2 | 32.6 | 19.3 | 96 | 75 | 64 |
| IR | 17.823 | 16.431 | 15.667 | 356.1 | 221.8 | 142.6 | 526 | 342 | 246 |
| BIR | 18.176 | 17.041 | 16.356 | 51.4 | 35.1 | 16.8 | 104 | 81 | 62 |




(a) Path length(m)




(b) Running time(ms)




(c) Number of sampling points

**Fig. 22.** Simulation data comparison graph in environment A.

In Figure 21(c), the Informed-RRT* algorithm is the most difficult to find the initial paths, and the number of iterations is the highest because its sampling range covers the whole three-dimensional space, and the algorithm is a unidirectional random tree. This results in the generated trajectories containing more folds but shorter path lengths. In contrast, the Bi-RRT* and Bi-Informed-RRT* algorithms are shown in Figure 21(b) and (d) have a faster expansion of the random tree when planning paths, but generates longer and less smooth paths.

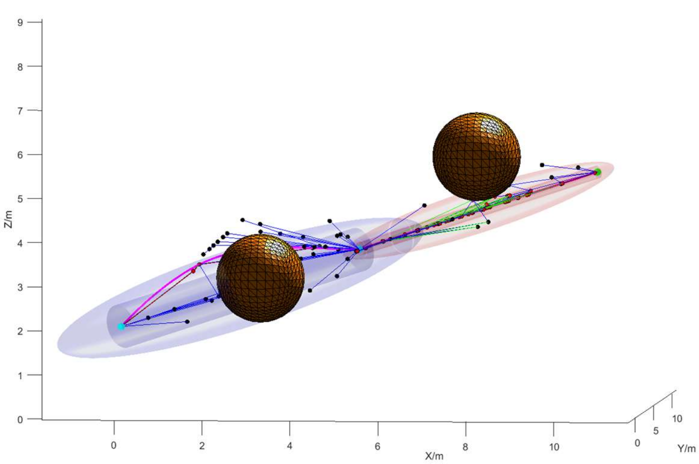
In Figure 21(a), the BPD-APF-Informed-FARRT* algorithm expands in parallel through four randomized trees. Instead of randomly sampling in the entire three-dimensional space, the algorithm employs a partition-biased sampling strategy to limit the sampling range, improving random tree growth's efficiency. APF is introduced in the generation process of new nodes; on the one hand, through repulsive force, the new nodes can effectively avoid obstacles to maintain a safe distance between the UAV and the obstacles. On the other hand, the gravitational force, makes the generation of new nodes more oriented and reduces the blindness of path exploration. In environment A, the environment consists of seven spheres and two cylinders, forming a scene full of obstacles. In this environment, the BPD-APF-Informed-FARRT* algorithm can utilize the adaptive step coefficients $\mu$ output from the fuzzy control system for fine tuning. In the dense obstacle region, due to the small adaptive step size coefficient output from the fuzzy control system, the movement step size of the new node is reduced accordingly, which allows the algorithm to perform a detailed search near the dense obstacles. On the contrary, in the open area away from the obstacles, the fuzzy control system outputs larger adaptive step coefficients, increasing the new node's step size and significantly improving the search efficiency. As shown in Figure 21(a), the distribution of nodes in regions with dense obstacles is denser, while nodes in open spaces are relatively sparse. In addition, the generated paths are smoother, and the yaw and pitch angles are controlled within the safe range, which ensures the flight safety of the UAV.

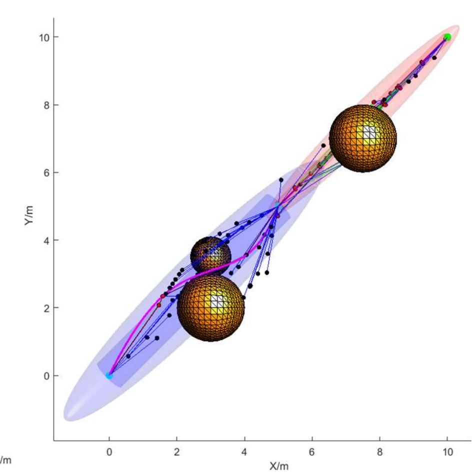
As can be seen from Figure 22 and Table 2, the BPD-APF-Informed-FARRT* algorithm plans outperform the other three algorithms in terms of average initial path length, average planning time, and average number of sampling points.

Compared with the Bi-RRT* algorithm, the improved algorithm reduced the average initial path length by 11.4%, the average planning time by 71.1%, and the average number of sampling points by 42.7%. Compared with the Informed-RRT* algorithm, the improved algorithm reduced the average initial path length by 8.9%, the average planning time by 95.8%, and the average number of sampling points by 87.4%. Compared to the Bi-Informed-RRT* algorithm, the improved algorithm has a 12.2% reduction in average initial path length, a 73.2% reduction in average planning time, and a 46.9% reduction in average number of sampling points. The improved algorithm can plan a better path with less iterations and planning time. The superiority of the BPD-APF-Informed-FARRT* algorithm is fully demonstrated.

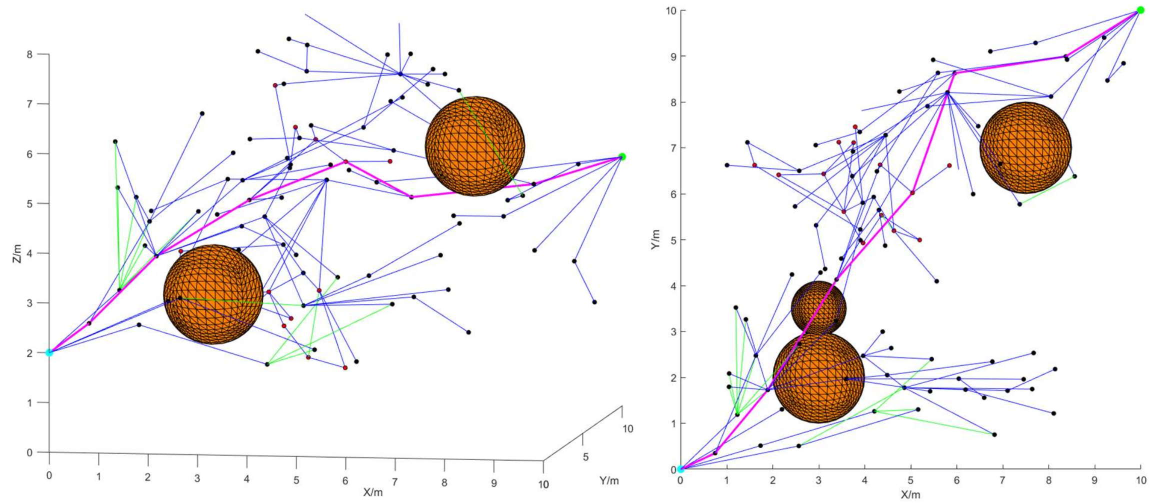
### 5.2. *Environment B*

In Environment B, a simple obstacle environment is set up. The path optimization results of the BPD-APF-Informed-FARRT* algorithm are compared with the Bi-RRT*, Informed-RRT*, and Bi-Informed-RRT* algorithms. Path planning experiments were conducted for 100, 200, 300, and 400 iterations for each algorithm (The Informed-RRT* algorithm in Environment B is not included in the comparison because the Informed-RRT* algorithm has a minimal probability of finding a path for the algorithm at several iterations less than 300.). Each set of iterations is run under environment B for 30 times respectively. The path planning results of the four algorithms at 100 iterations are shown in Figure 23.

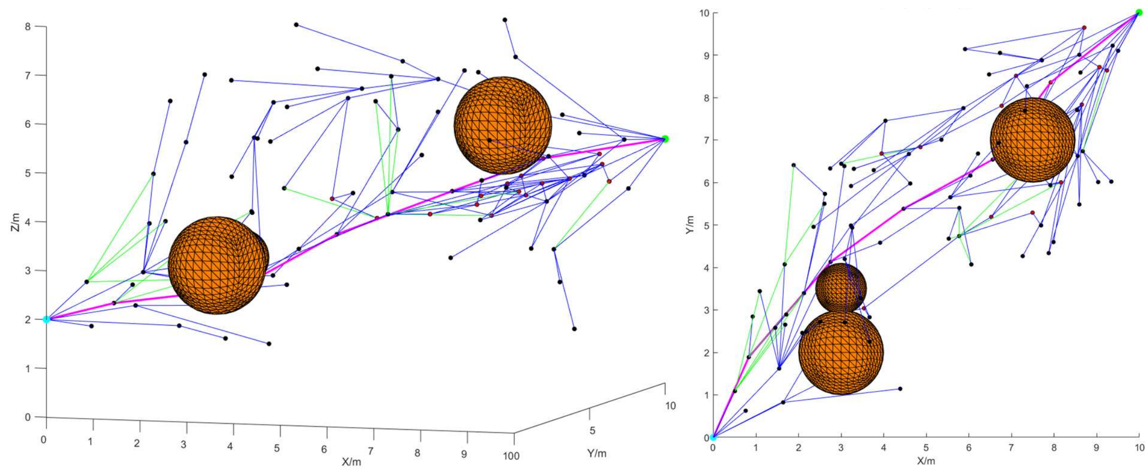



(a) Front and top views of the BAIFR algorithm path planning




(b) Front and top views of the BR algorithm path planning




(c) Front and top views of the BIR algorithm path planning

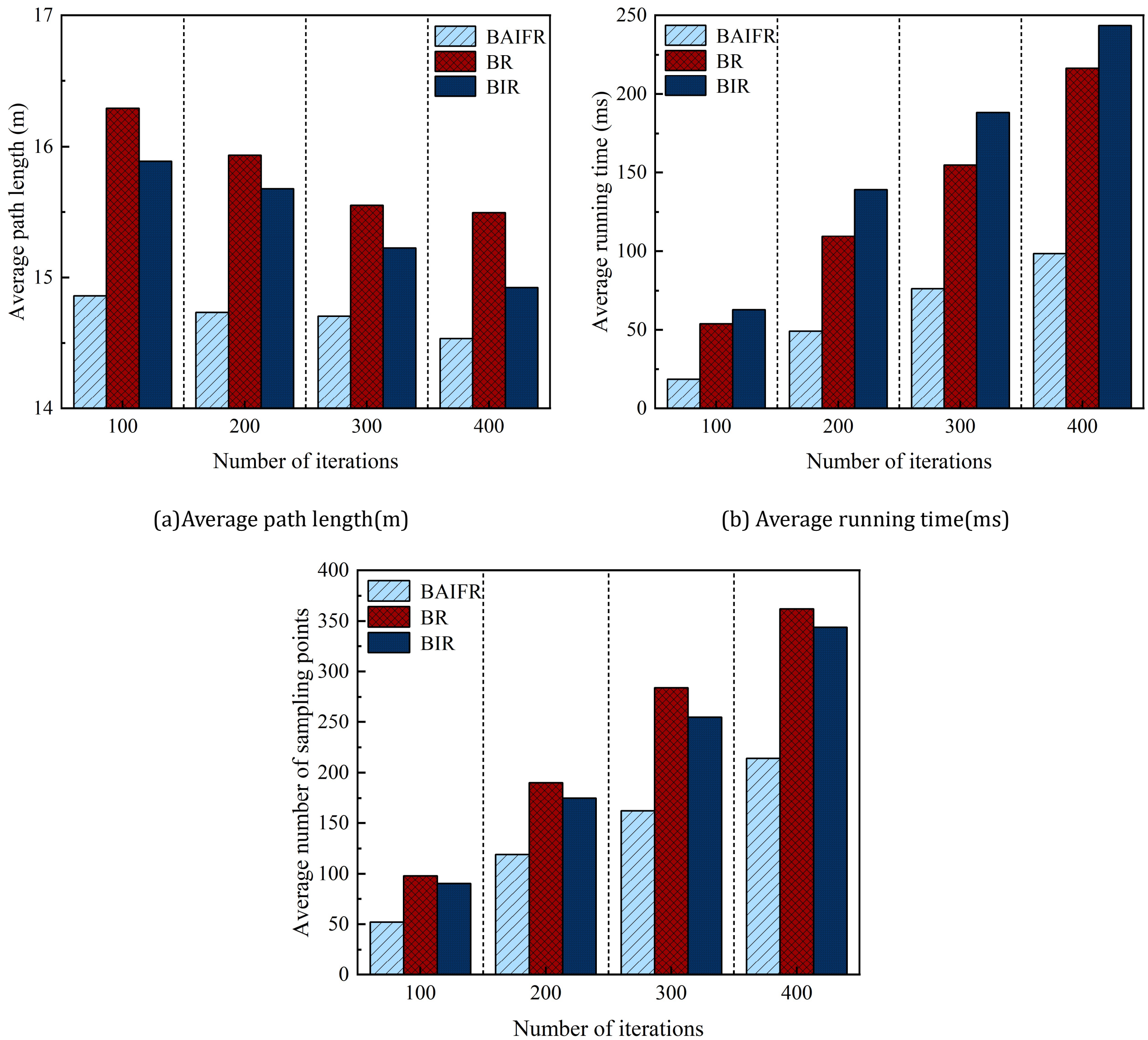
**Fig. 23.** Path planning effect at 100 iterations in environment B.

Each algorithm was run independently for 30 times for 100, 200, 300, and 400 iterations of path planning experiments under the same conditions. Comparative experimental results are shown in Figure 24 and Table 3.

**Table 3.** Simulation date in environment B.

| Number of iterations | Algorithm | Average path length (m) | Average running time (ms) | Average number of sampling points |
|---|---|---|---|---|
| 100 | BAIFR | 14.861 | 18.7 | 52 |
| | BR | 16.291 | 53.8 | 98 |
| | BIR | 15.885 | 62.9 | 90 |
| 200 | BAIFR | 14.736 | 49.1 | 119 |
| | BR | 15.934 | 109.6 | 190 |
| | BIR | 15.677 | 139.2 | 175 |
| 300 | BAIFR | 14.707 | 76.3 | 162 |
| | BR | 15.552 | 154.8 | 284 |
| | BIR | 15.223 | 188.3 | 255 |

| | | | | |
|---|---|---|---|---|
| 400 | BAIFR | 14.533 | 98.5 | 214 |
| | BR | 15.496 | 216.2 | 362 |
| | BIR | 14.924 | 241.7 | 344 |




(a)Average path length(m)

(b) Average running time(ms)

(c) Average number of sampling points

**Fig. 24.** Simulation data comparison graph in environment B.

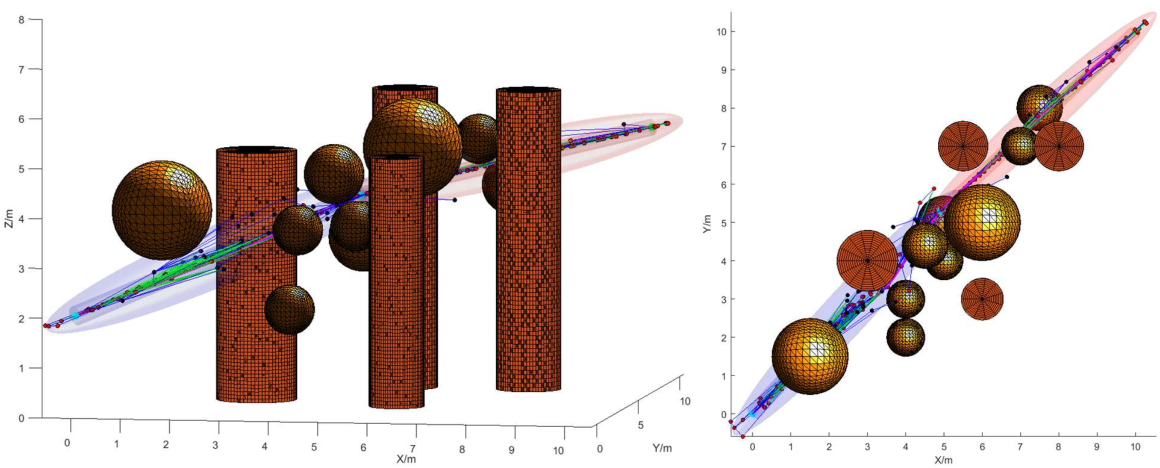
The trajectories planned by the three algorithms in Environment B are shown in Figure 23. The trajectory of the new algorithm in this paper is shown in Figure 23(a). The BPD-APF-Informed-FARRT* path planning algorithm can be rapidly restricted to smaller ellipsoidal domains due to the short initial paths planned, and the fusion of multiple strategies for improvement, which results in the improved algorithms generating fewer redundancy points, and the trajectories are shorter, straighter, and smoother, which better meets the requirements of the flight of UAVs.

By analyzing the data in Figure 24 and Table 3, the BPD-APF-Informed-FARRT* algorithm outperforms the other three algorithms in terms of average path length, average planning time, and average number of sampling points planned for different numbers of iterations.

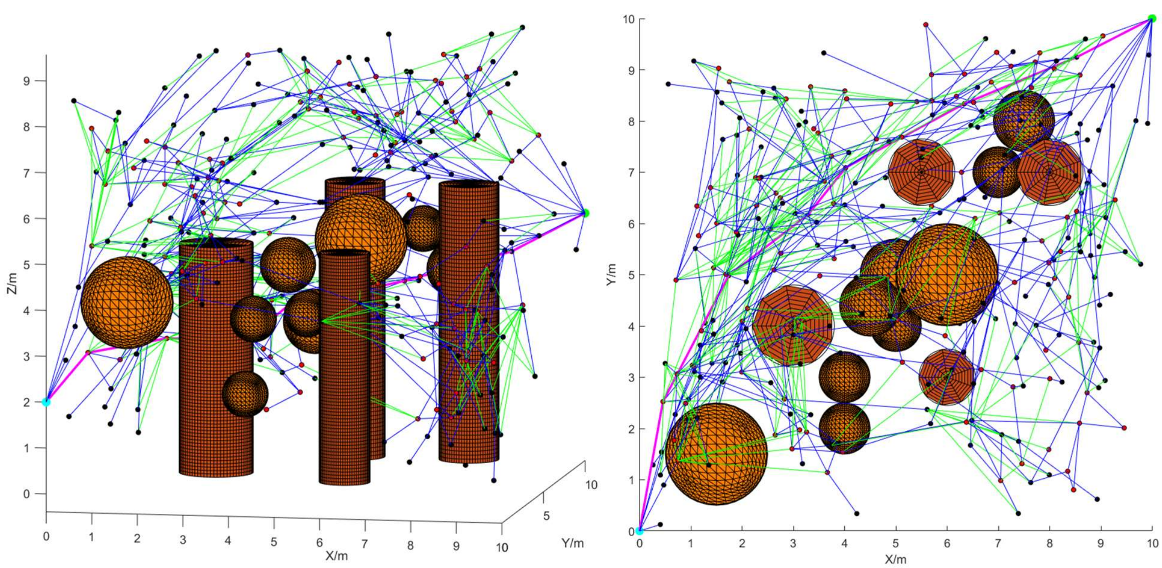
The number of iterations is 400 as an example. Compared with the Bi-RRT* algorithm, the improved algorithm has a 6.2% shorter average path length, 54.4% shorter average planning time, and a 40.9% fewer average number of sampling points. Compared with the Bi-Informed-RRT* algorithm, the improved algorithm has a 2.6% reduction in average path length, a 59.2% reduction in average planning time, and a 37.7% reduction in average number of sampling points.

### *5.3. Environment C*

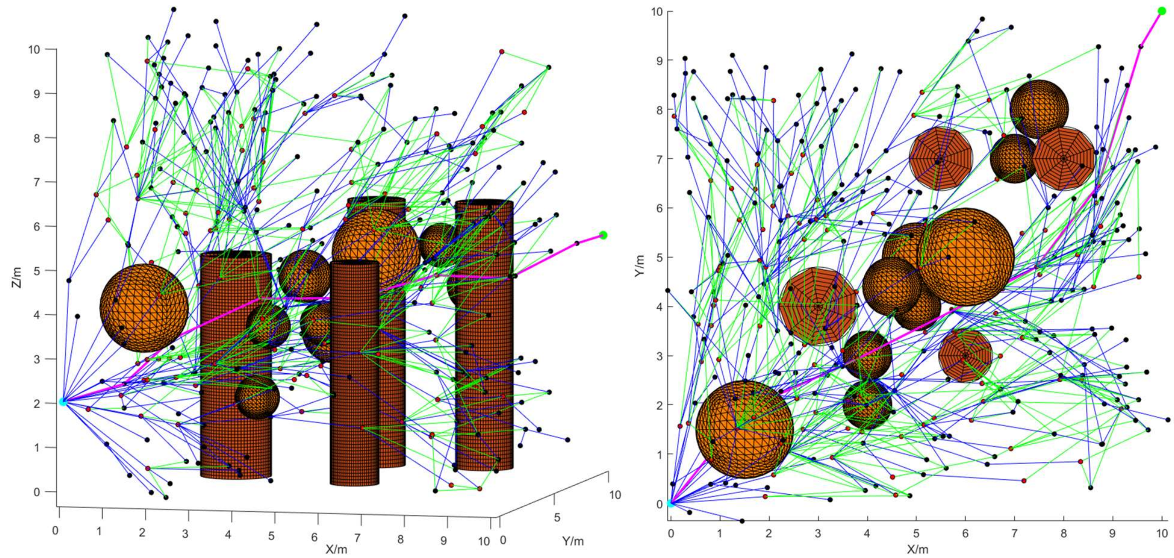
In Environment C, a complex obstacle environment is set up. The path optimization results of the BPD-APF-Informed-FARRT* algorithm are compared with the Bi-RRT*, Informed-RRT*, and Bi-Informed-RRT* algorithms. Path planning experiments were conducted for 400, 500, 600, and 700 iterations for each algorithm. Each set of iterations is run under environment C 30 times. The path planning results of the four algorithms at 400 iterations are shown in Figure 25.



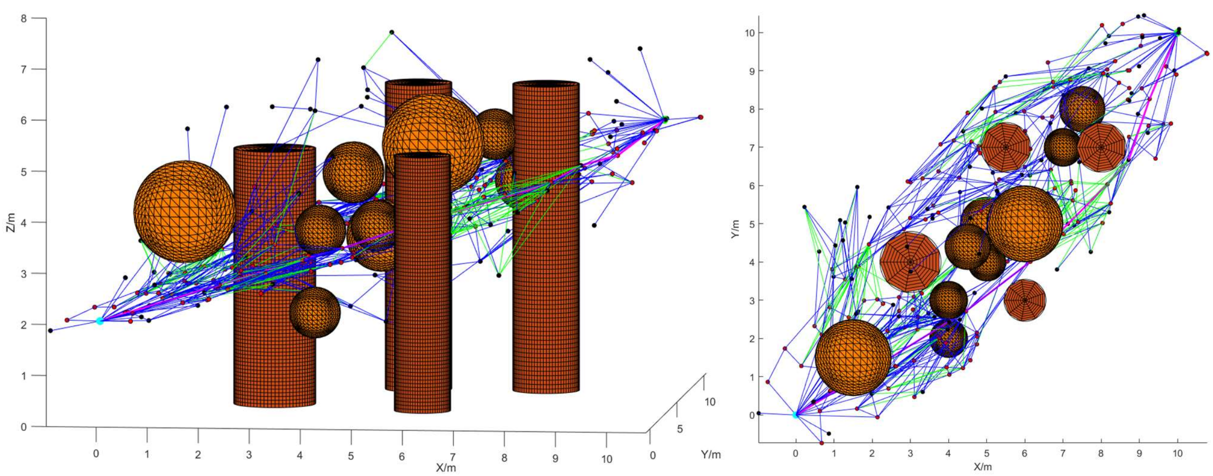

(a) Front and top views of the BAIFR algorithm path planning




(b) Front and top views of the BR algorithm path planning




(c) Front and top views of the IR algorithm path planning



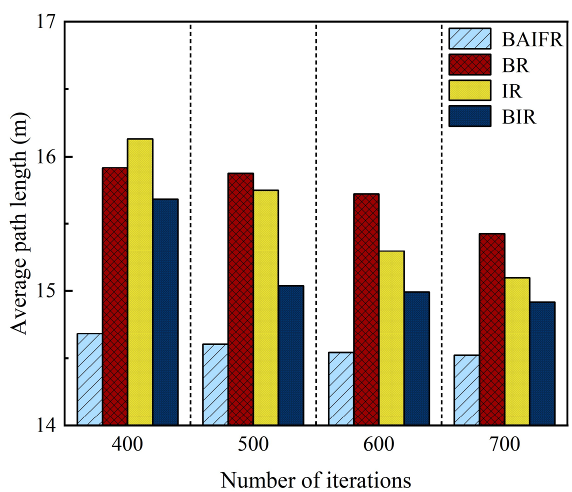

(d) Front and top views of the BIR algorithm path planning

**Fig. 25.** Path planning effect at 400 iterations in environment C.

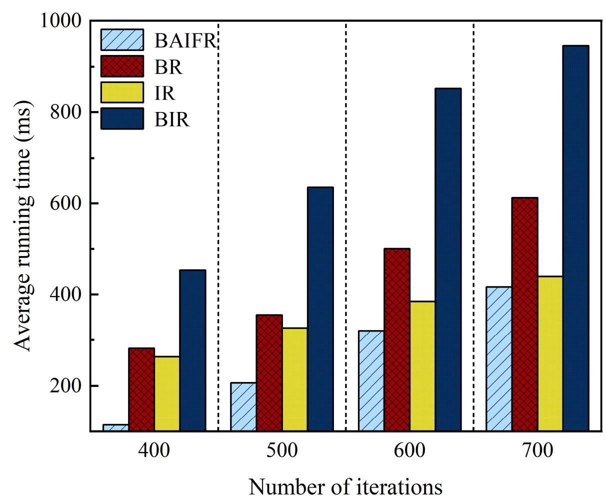
Each algorithm was run independently for 30 times for 400, 500, 600, and 700 iterations of path planning experiments under the same conditions. Comparative experimental results are shown in Figure 26 and Table 4.

**Table 4.** Simulation date in environment C.

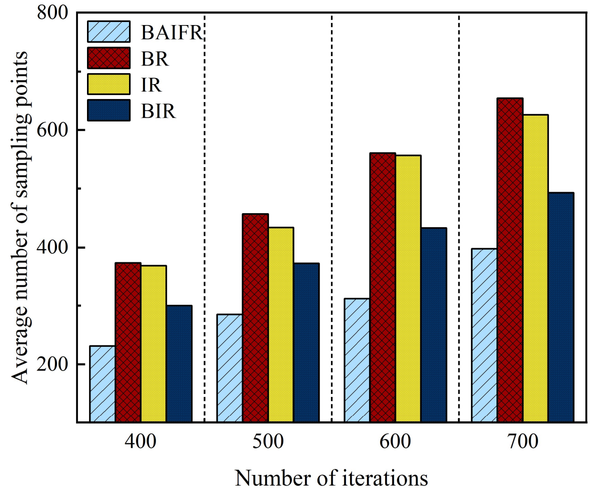
| Number of iterations | Algorithm | Average path length (m) | Average running time (ms) | Average number of sampling points |
|---|---|---|---|---|
| 400 | BAIFR | 14.682 | 110.6 | 231 |
| | BR | 15.917 | 281.9 | 373 |
| | IR | 16.132 | 273.7 | 368 |
| | BIR | 15.794 | 453.1 | 300 |
| 500 | BAIFR | 14.602 | 206.4 | 285 |
| | BR | 15.874 | 355.3 | 457 |
| | IR | 15.748 | 326.8 | 434 |
| | BIR | 15.037 | 635.1 | 372 |
| 600 | BAIFR | 14.541 | 320.8 | 312 |
| | BR | 15.721 | 500.6 | 561 |
| | IR | 15.295 | 384.7 | 557 |
| | BIR | 14.992 | 851.5 | 433 |
| 700 | BAIFR | 14.522 | 416.3 | 398 |
| | BR | 15.426 | 612.1 | 654 |
| | IR | 15.096 | 439.1 | 626 |
| | BIR | 14.916 | 945.6 | 493 |




(a)Average path length(m)




(b) Average running time(ms)




(c) Average number of sampling points

**Fig. 26.** Simulation data comparison graph in environment C.

The advantages of the BPD-APF-Informed-FARRT* algorithm are more evident in complex obstacle environments. As can be seen in Figure 25(a), in the area with fewer obstacles in the lower left and upper right corners, the improved BPD-APF-Informed-FARRT* algorithm uses larger step sizes for expansion. The number of sample points in this area is significantly lower. In contrast, in the densely obstacle-filled area in the center of the map, the step size is relatively small, so the complex environment can be searched carefully to reduce the length of the planned path. Compared to the other three algorithms in Figure 25, the BPD-APF-Informed-FARRT* algorithm focuses on optimizing the sampling in the area around the optimal path, resulting in a smoother final path and a lower path cost. This algorithm implements a simultaneous search of the four trees. The search time is significantly reduced, and many redundant points are reduced.

Data in Table 4 are analyzed, taking 400 iterations as an example. Regarding average path length, the improved algorithm is 7.1% shorter than the suboptimal BIR algorithm. Regarding average running time, the improved algorithm is 59.6% shorter than the suboptimal IR algorithm. Regarding the average number of sampling points, the improved algorithm is 23% shorter than the suboptimal BIR algorithm.

Compared with the simple obstacle data, it can be found that the improved algorithm has more advantages regarding path length and running time in the complex obstacle environment.

## 6. Conclusions

An improved BPD-APF-Informed-FARRT* algorithm is proposed in this paper to overcome the limitations of RRT*, such as poor target bias, slow convergence, and tortuous paths exist. This improved algorithm enhances performance by introducing multiple strategies.

In order to verify the effectiveness of the improved algorithm, simulation experiments of initial path generation and path optimization were carried out for the four algorithms in three map environments. The results show that the BPD-APF-Informed-FARRT* algorithm significantly reduces the time required for path planning while constraining the yaw and pitch angles and generating a shorter and smoother path. The path planned by this algorithm is more suitable for the UAV to fly in real-world environments.

This improved algorithm has yet to be applied in real-world environments, and its effectiveness will be verified in subsequent path planning scenarios for real drones. In addition, for complex environments, we need to combine global path planning with local path planning strategies, which is also the focus of subsequent research in this paper.

## Funding

This work was supported by Yunnan Horizontal Science and Technology Service Program KKF0202303309.

Junli Li was born in Kunming, Yunnan. China, in 1974. She obtained a master's degree in engineering from Kunming University of Science and Technology in China. She is currently engaged in teaching and research at the School of Information Engineering and Automation, Kunming University of Science and Technology. Her main focus is on teaching and research in the area of intelligent control algorithms.

## References

[1] Karaca, Y., Cicek, M., Tatli, O., Sahin, A., Pasli, S., Beser, M. F., & Turedi, S. (2018). The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. The American journal of emergency medicine, 36(4), 583-588.

[2] Abbas, A., Zhang, Z., Zheng, H., Alami, M. M., Alrefaei, A. F., Abbas, Q., ... & Zhou, L. (2023). Drones in plant disease assessment, efficient monitoring, and detection: A way forward to smart agriculture. Agronomy, 13(6), 1524.

[3] Picchio, F., Parrinello, S., & Barba, S. (2022). Drones and Drawings-methods of data acquisition, management, and representation. Disegnarecon, 15(29), 1-7.

[4] Khan, M. T. R., Muhammad Saad, M., Ru, Y., Seo, J., & Kim, D. (2021). Aspects of unmanned aerial vehicles path planning: Overview and applications. International Journal of Communication Systems, 34(10), e4827.

[5] Wu, J., Wang, H., Li, N., Yao, P., Huang, Y., & Yang, H. (2018). Path planning for solar-powered UAV in urban environment. Neurocomputing, 275, 2055-2065.

[6] Aggarwal, S., & Kumar, N. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. Computer communications, 149, 270-299.

[7] Zhao, Y., Zheng, Z., & Liu, Y. (2018). Survey on computational-intelligence-based UAV path planning. Knowledge-Based Systems, 158, 54-64.

[8] Ait Saadi, A., Soukane, A., Meraihi, Y., Benmessaoud Gabis, A., Mirjalili, S., & Ramdane-Cherif, A. (2022). UAV path planning using optimization approaches: A survey. Archives of Computational Methods in Engineering, 29(6), 4233-4284.

[9] Puente-Castro, A., Rivero, D., Pazos, A., & Fernandez-Blanco, E. (2022). A review of artificial intelligence applied to path planning in UAV swarms. Neural Computing and Applications, 34(1), 153-170.

[10] Mardani, A., Chiaberge, M., & Giaccone, P. (2019). Communication-aware UAV path planning. IEEE Access, 7, 52609-52621.

[11] Bashir, N., Boudjit, S., Dauphin, G., & Zeadally, S. (2023). An obstacle avoidance approach for UAV path planning. Simulation modelling practice and theory, 129, 102815.

[12] Lin, Y., & Saripalli, S. (2017). Sampling-based path planning for UAV collision avoidance. IEEE Transactions on Intelligent Transportation Systems, 18(11), 3179-3192.

[13] Cabreira, T. M., Brisolara, L. B., & Paulo R, F. J. (2019). Survey on coverage path planning with unmanned aerial vehicles. Drones, 3(1), 4.

[14] Luo, J., Wang, Z., Xia, M., Wu, L., Tian, Y., & Chen, Y. (2023). Path planning for UAV communication networks: Related technologies, solutions, and opportunities. ACM Computing Surveys, 55(9), 1-37.

[15] Maboudi, M., Homaei, M., Song, S., Malihi, S., Saadatseresht, M., & Gerke, M. (2023). A review on viewpoints and path planning for UAV-based 3-D reconstruction. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 16, 5026-5048.

[16] Fu, Z., Yu, J., Xie, G., Chen, Y., & Mao, Y. (2018). A heuristic evolutionary algorithm of UAV path planning. Wireless Communications and Mobile Computing, 2018(1), 2851964.

[17] Bayerlein, H., Theile, M., Caccamo, M., & Gesbert, D. (2021). Multi-UAV path planning for wireless data harvesting with deep reinforcement learning. IEEE Open Journal of the Communications Society, 2, 1171-1187.

[18] Wang, X., Gursoy, M. C., Erpek, T., & Sagduyu, Y. E. (2022). Learning-based UAV path planning for data collection with integrated collision avoidance. IEEE Internet of Things Journal, 9(17), 16663-16676.

[19] Cui, Z., & Wang, Y. (2021). UAV path planning based on multi-layer reinforcement learning technique. Ieee Access, 9, 59486-59497.

[20] Khachumov, M. (2019, August). A Rule-based approach for controlling UAVs formation flight. In Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings" ER (ZR) 2019, Kursk, Russia, 17-20 April 2019 (pp. 319-330). Singapore: Springer Singapore.

[21] Hu, J., Wang, T., Zhang, H., Pan, Q., Zhang, J., & Xu, Z. (2023). A review of rule-based collision avoidance technology for autonomous UAV. Science China Technological Sciences, 66(9), 2481-2499.

[22] Bai, Z., Shi, J., Li, Z., Li, M., & Chen, K. C. (2024). Rule-Guided DRL for UAV-Assisted Wireless Sensor Networks With No-Fly Zones Safety. IEEE Transactions on Cognitive Communications and Networking.

[23] Noreen, I., Khan, A., & Habib, Z. (2016). Optimal path planning using RRT* based approaches: a survey and future directions. International Journal of Advanced Computer Science and Applications, 7(11).

[24] Chai, X., Zheng, Z., Xiao, J., Yan, L., Qu, B., Wen, P., ... & Sun, H. (2022). Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment. Aerospace Science and Technology, 121, 107287.

[25] Wang, J., Li, Y., Li, R., Chen, H., & Chu, K. (2022). Trajectory planning for UAV navigation in dynamic environments with matrix alignment Dijkstra. Soft Computing, 26(22), 12599-12610.

[26] Chen, J., Li, M., Yuan, Z., & Gu, Q. (2020, June). An improved A* algorithm for UAV path planning problems. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (Vol. 1, pp. 958-962). IEEE.

[27] Xu, Z., Deng, D., & Shimada, K. (2021). Autonomous UAV exploration of dynamic environments via incremental sampling and probabilistic roadmap. IEEE Robotics and Automation Letters, 6(2), 2729-2736.

[28] Pan, Z., Zhang, C., Xia, Y., Xiong, H., & Shao, X. (2021). An improved artificial potential field method for path planning and formation control of the multi-UAV systems. IEEE Transactions on Circuits and Systems II: Express Briefs, 69(3), 1129-1133.

[29] Wanga, H., Liu, X., Song, S., Li, B., Lu, X., Nie, J., & Zhao, X. (2021, December). Improved RRT path planning algorithm based on growth evaluation. In International Conference on Intelligent Equipment and Special Robots (ICIESR 2021) (Vol. 12127, pp. 521-527). SPIE.

[30] Cekmez, U., Ozsiginan, M., & Sahingoz, O. K. (2016, June). Multi colony ant optimization for UAV path planning with obstacle avoidance. In 2016 international conference on unmanned aircraft systems (ICUAS) (pp. 47-52). IEEE.

[31] Silva Arantes, J. D., Silva Arantes, M. D., Motta Toledo, C. F., Júnior, O. T., & Williams, B. C. (2017). Heuristic and genetic algorithm approaches for UAV path planning under critical situation. International Journal on Artificial Intelligence Tools, 26(01), 1760008.

[32] He, Y., & Wang, M. (2024). An improved chaos sparrow search algorithm for UAV path planning. Scientific reports, 14(1), 366.

[33] Fan, J., Chen, X., & Liang, X. (2023). UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased. Expert systems with applications, 213, 119137.

[34] Guo, Y., Liu, X., Liu, X., Yang, Y., & Zhang, W. (2022). FC-RRT*: An improved path planning algorithm for UAV in 3D complex environment. ISPRS International Journal of Geo-Information, 11(2), 112.

[35] Guo, J., Xia, W., Hu, X., & Ma, H. (2022). Feedback RRT* algorithm for UAV path planning in a hostile environment. Computers & Industrial Engineering, 174, 108771.

[36] Fusic, S. J., & Sitharthan, R. (2024). Improved RRT* algorithm-based path planning for unmanned aerial vehicle in a 3D metropolitan environment. Unmanned Systems, 12(05), 859-875.

[37] Wu, X., Xu, L., Zhen, R., & Wu, X. (2019). Biased sampling potentially guided intelligent bidirectional RRT∗ algorithm for UAV path planning in 3D environment. Mathematical problems in engineering, 2019(1), 5157403.