Hindawi
International Journal of Aerospace Engineering
Volume 2022, Article ID 1166968, 14 pages
https://doi.org/10.1155/2022/1166968

Research Article

Improved F-RRT* Algorithm for Flight-Path Optimization in Hazardous Weather

Xue Qiu, Yaohui Li, Rui Jin, Zhi Zhao, Jiajun Li, Donglin Lu, and Linhui Ma

Air Traffic Management College, Civil Aviation Flight University of China, Guanghan, Sichuan 618307, China

Correspondence should be addressed to Xue Qiu; zhaozhi@cafuc.edu.cn

Received 1 April 2022; Revised 9 July 2022; Accepted 15 July 2022; Published 22 August 2022

Academic Editor: Jinchao Chen

Hazardous weather has become a major cause of flight delays in recent years. With the development of satellite navigation systems, the study of flight-path optimization under hazardous weather conditions has become especially important. In this study, radar data were used as the basis for the initial flight-restricted area under hazardous weather conditions, and the Graham algorithm was used to delineate the dynamic flight-restricted area by comprehensively considering the hazardous weather boundary changes along with the speed and direction. Then, under the grid environment model, the range of influence, size, and distribution characteristics of the flight-restricted area was examined, and the path optimization model was created according to constraints related to the path distance, corner size, and number of turning points. An improved F-RRT* algorithm was developed to solve the model. The algorithm can overcome the problems of traditional path planning algorithms, such as strong randomness, poor guidance, slow convergence speed, unsmooth paths, and poor tracing smoothness. Finally, a simulation analysis was conducted on the Guiyang–Guangzhou route in China as an example. This study can address the drawbacks of existing research on route change and provide sufficient theoretical support and reference for the implementation of specific route change plans in the future.

1. Introduction

Hazardous weather, which primarily refers to small and medium-scale weather elements such as thunderstorms, squall lines, icing, and wind shear [1], has become a major threat to air transport safety and causes flight delays. In recent years, it has accounted for approximately 50% of the total flight accidents. It is necessary to identify hazardous weather accurately and provide the best route during aircraft operation, which is of great significance to ensure the safety of aircraft navigation [2].

At present, diversion models are generally used to determine the optimal route under hazardous weather, and several studies have been reported. Ahlstrom and Jaggard evaluated the implications of hazardous weather on flight operations and proposed a method to identify dangerous weather during flight [3]. Liu et al. comprehensively studied hazardous weather, FFA, aircraft, and the flight coordination of aircraft and established an aircraft diversion decision model in extreme weather [4]. Chen et al. studied the coverage path planning problem of autonomous heterogeneous unmanned aerial vehicles (UAVs) over a limited number of areas and realized a flight path covering all the areas of interest [5]. Ding and Sui comprehensively analyzed the damage of route-based systems on meteorological data, identified the flight segments that pilots should avoid in route selection, and then used an A* algorithm to divert the route [6]. Minyang et al., to solve the path planning problem of multi-degree-of-freedom robots in complex environments, proposed an RRT* algorithm that can address the frequent demand for UAVs [7]. Song et al. introduced several representative methods of path planning in the field of UAVs and provided several parameters for the 3D flight path planning of UAVs [8]. Chen et al. focused on the coverage path planning problem of heterogeneous UAVs and proposed an ant-colony-system-based algorithm to select appropriate UAV paths and effectively cover all areas [9].

However, most current studies focus on the diversion of UAVs [10] and a few on the diversion of flights in hazardous weather, as found that most researchers focused on the route

planning of flight diversions [11] with not fully consider the implications of dangerous weather levels and movement trends on the routes. To address these limitations, this study delineates and predicts dangerous weather areas, utilizes these areas as constraints, evaluates other limiting factors such as aircraft performance and flight safety, and establishes a diversion path. Further, a mathematical model of route planning is realized. In addition, this study proposes an improved F-RRT* algorithm based on the goal-biased strategy [12]. Most researchers used traditional path planning algorithms, such as A* [13], genetic [14], simulated annealing [15], and ant colony optimization algorithms, to realize flight diversion models [16]. However, the computational complexity of these algorithms is exponentially related to the degrees of freedom of the aircraft. Therefore, based on a previous study [17], this study proposes an improved F-RRT* algorithm based on a target bias strategy combined with the angle limit. It has demonstrated the ability to determine safe trajectories between two states in high-dimensional problems using system dynamics and constraints. Unlike other algorithms, the RRT* algorithm does not require an initial guess because it can determine a feasible path in a few iterations and then optimize the path in subsequent steps [18]. Initially, the traditional RRT* algorithm was used for robot path planning. Recently, it has been employed to study UAVs [19]. However, it has not been successfully applied for flight path planning. To address this limitation, this study modifies an RRT* algorithm according to the characteristics of the flight and obtains an F-RRT* algorithm, which is suitable for flight path optimization. This approach solves the problems of strong randomness, poor guidance, slow convergence, tortuous paths, and the poor tracking smoothness of traditional path planning algorithms [20]. It can more effectively solve the problem of flight path planning in dangerous weather.

2. Delineation of Flight-Restricted Areas in Hazardous Weather

Hazardous weather during flights mainly refers to hail, heavy rain, and thunderstorms. These hazardous weather conditions have adverse effects on flight safety and normal operation of the airspace system [21]. Reasonable and accurate delineation of the flight-restricted area is essential to reduce these adverse effects. In this study, we mainly used the monitored radar echo data to filter and divide the data in order to determine the initial range of the flight-restricted area [22]. The Graham algorithm was then used for the dynamic prediction of the flight-restricted area.

2.1. Initial Designation of Flight-Restricted Area in Hazardous Weather. In this study, the flight-restricted area was determined based on the radar echo map. First, the area of radar weather echo intensity ≥ 41 dBZ was determined as the dangerous flight area according to the characteristics of heavy rainfall [23]. Then, image processing techniques were used to identify the hazardous weather areas in grayscale. Finally, the outline of the flight-restricted area was circled using convex polygons. The steps to delineate the flight-restricted area are as follows.

(1) Obtain the relevant hazardous weather data, extract the areas with radar weather echo intensity ≥ 41 dBZ, and calculate the radar reflectance. Each radar reflectance corresponds to an RGB color value, and the area constituted by the pixel points combining these color values is the hazardous weather area

(2) Binarize the images of severe weather areas, thus highlighting them and making the background grayer

(3) Delineate the initial outline of the severe weather area. The binary image obtained in the second step is further recognized to improve the rough outline of the severe weather area

(4) Approximate the contour lines using irregular polygons. In this step, the resulting contour set is approximated using irregular polygons of a given accuracy, and a simple polygonal contour set is created after approximation

(5) Create a convex polygon. The convex packets of each simple polygon are located; the area enclosed by these convex polygons is the flight-restricted area. Subsequently, the delineated danger zone is processed

   (a) Denoising process: according to the flight performance database, a danger zone of less than seven miles can be neglected when the aircraft is flying at a speed of no greater than 420 kt

   (b) Combined processing: according to civil aviation regulations, the minimum separation between clouds should be at least 20 km for the aircraft to pass through them; clouds that do not meet the above condition are considered as a whole [24]

2.2. Dynamic Prediction of Flight-Restricted Area in Hazardous Weather. To perform the dynamic prediction of the flight-restricted area under hazardous weather, we first need to determine the initial convex polygon formed by the hazardous weather boundary area. Then, we introduce the gray prediction model, combine the data before the hazardous weather boundary point, and use the Graham algorithm to dynamically predict the changes in the hazardous weather boundary point within a short period of time in the future. Finally, the predicted area is extended outward by 25 km to determine the extent of the flight-restricted area at a certain time in the future.

The algorithm first scans the planar point set to find the extreme points in the leftmost, rightmost, topmost, and bottommost directions to construct an initial convex packet and delete all the points inside the initial convex packet. Then, it groups the remaining point sets and generates a new convex packet for each group by using the Graham algorithm. Finally, the vertices of all subsets of the convex packets are considered

as a new point set, and the Graham algorithm is applied again to generate the final convex packet.

Based on the above analysis, the Graham algorithm is divided into the following three steps. (1) The individual reference points on the graph are determined. (2) The polar angles are extended outward from each reference point, and the resulting polar angles are sorted to form a simple polygon. (3) The concave points on the simple polygon are removed, and the individual convex points are then joined to form a convex polygon. The time complexity of the convex packet algorithm is largely determined by that of the sorting algorithm. Because the time complexity of (1) and (3) is $O(n)$, that of (2) is $O(n \log n)$. Therefore, the total time complexity is $O(n \log n)$. At this point, the speed of computing the convex packets is relatively slow. Therefore, if the time complexity of the sorting algorithm can be reduced, the speed of computing the convex packets can be improved significantly [25]. The process is illustrated in Figure 1 below.

2.3. Movement Prediction Model of Flight-Restricted Area. Previous studies proposed dynamic predictions for light-prohibited areas under the assumption that the moving speed and direction of extreme weather are known. However, in practical applications, these parameters cannot easily be determined. Moreover, the relationship between the movement of a flight-restricted area along the x and y directions and time can be studied based only on the constantly updated meteorological radar data. Subsequently, the movement prediction model of this area can be established.

As the update frequency of meteorological radar data used in this study is 5 min, the starting time t_n of the forecast is used as the benchmark. By converting the forecast time $t_i (i = 1, 2, 3, \cdots)$ into the forecast time slice $T_i (T_i = 1, 2, 3, \cdots)$, the conversion formula can be expressed as

$$T_i = \left[\frac{(t_i - t_n)}{5}\right]. \tag{1}$$

2.3.1. Prediction Model of Uniform Movement. The motion of a thunderstorm can be modeled as a uniform motion along a certain direction in a short time as its component speeds along the x and y directions are unchanged. Therefore, the centroid positions of a thunderstorm, x_i and y_i, correspond to n historical time slices $T_i (T_i = 1, 2, 3, \cdots)$. Using the relationships among the speed, distance, and time, the moving speeds v_x and v_y along the x and y directions, respectively, are obtained as follows:

$$\begin{aligned} v_x &= \frac{\sum_{i=0}^{n-1} (x_{i+1} - x_i)}{n-1}, \\ v_y &= \frac{\sum_{i=0}^{n-1} (y_{i+1} - y_i)}{n-1}. \end{aligned} \tag{2}$$

According to the moving speeds, v_x and v_y, and the thunderstorm positions, x_n and y_n, corresponding to the n time slices Tn, we can predict the position of the thunderstorm after $i (i = 1, 2, 3, \cdots n)$ time slices as follows:

$$\begin{aligned} x_{n+i} &= x_n + v_x * i, \\ y_{n+i} &= y_n + v_y * i. \end{aligned} \tag{3}$$

2.3.2. Prediction Model of Uniform Speed Movement. In Section 2.3.1, the thunderstorm motion is regarded as a uniform motion. However, in summer, thunderstorms exhibit uniform acceleration or uniform deceleration during the development and dissipation phases. Therefore, according to the centroid positions of the thunderstorm, x_i and y_i, corresponding to n historical time slices $T_i (T_i = 1, 2, 3, \cdots n)$ and using the relationship between acceleration, speed, and time, the accelerations a_x and a_y along the x and y directions, respectively, are obtained as follows:

$$a_x = \frac{\sum_{i=0}^{n-2} \{(x_{i+2} - x_{i+1}) - (x_{i+1} - x_i)\}}{n-2}, \tag{4}$$

$$a_y = \frac{\sum_{i=0}^{n-2} \{(y_{i+2} - y_{i+1}) - (y_{i+1} - y_i)\}}{n-2}. \tag{5}$$

In Equations (4) and (5), $(x_{i+2} - x_{i+1}) - (x_{i+1} - x_i)$ and $(y_{i+2} - y_{i+1}) - (y_{i+1} - y_i)$ represent the velocity change rates of two adjacent time slices. By calculating the average accelerations a_x and a_y and using the thunderstorm positions x_n and y_n corresponding to the time slice T_n, the position of the thunderstorm after $i (i = 1, 2, 3, \cdots n)$ time slices can be expressed as follows:

$$\begin{aligned} x_{n+i} &= x_n + (x_n - x_{n-1}) * i + \frac{1}{2} a_x i^2, \\ y_{n+i} &= y_n + (y_n - y_{n-1}) * i + \frac{1}{2} a_y i^2. \end{aligned} \tag{6}$$

3. Multiobjective Redirected Path Optimization Model

3.1. Three Types of Flight-Restricted Areas. Considering the shape, size, and distribution characteristics of the flight-restricted areas, this study classified them into three main categories—block-shaped, strip-shaped, and scattered. A block-shaped flight-restricted area is a convex polygon with the horizontal scale of the flight-restricted area usually larger than $100 \times 100\ \text{km}^2$; the affected flight area is usually large [26]. Both dynamic flight-restricted areas caused by block clouds and those formed over a period of time are categorized as block-shaped flight-restricted areas.

There are generally two types of strip-shaped flight-restricted areas. The first is a single-strip flight-restricted area with a long and narrow convex polygon. The second is a series of small, banded flight-restricted areas with long and narrow strips distributed along both sides of the route. The strip-shaped flight restriction zone covers almost 2–3 points. The ratio between the long and short sides of the outer rectangle is greater than two.

Scattered flight-restricted areas refer to multiple small-scale flight-restricted areas along the route on both sides of the arbitrary distribution, which have the characteristics of

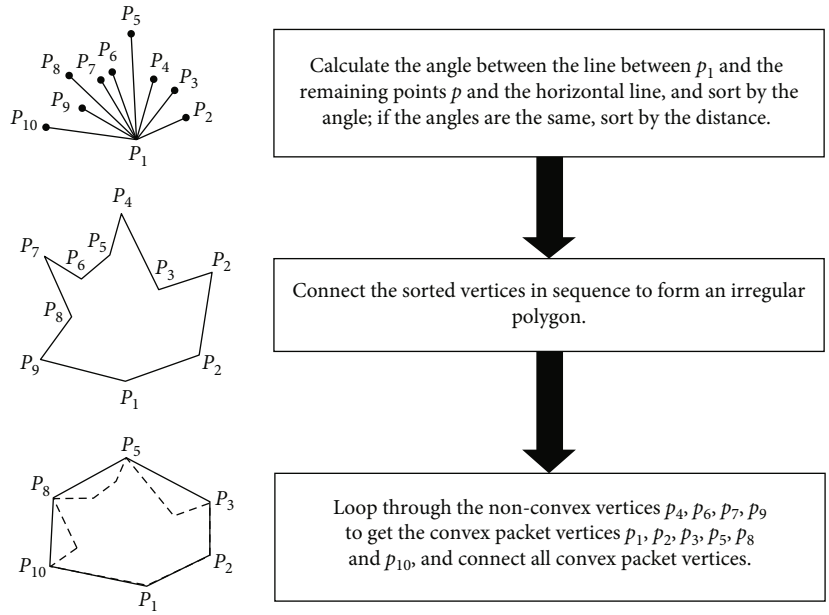

Figure 1: Construction of a convex packet using the Graham algorithm.

small scale, large number, and small impact range. The main difference between the scattered and block-shaped flight-restricted areas is that the latter is limited to a single flight-restricted area, whereas the former consists of multiple flight-restricted areas. The geometric scale of the latter is much smaller than that of the former. Unlike the strip-shaped distributed flight-restricted area, the scattered flight-restricted area is distributed among a small number of route points. The strip distribution is not obvious. Moreover, the distribution is irregular, and the scale of the flight-restricted area is small [27]. The distribution of banded flight-restricted areas covers almost the entire route, involving at least five waypoints with distinct banding characteristics. The dynamic flight-restricted areas caused by the scattered small-scale clouds and the flight-restricted areas formed during the fixed period belong to this category. The distribution of the scattered flight-restricted areas has no specific pattern, and the number of restricted areas may be large or small. Therefore, the grid approach has advantages of simplicity, intuitiveness, and ease of implementation.

(Abridged here)

3.2. Description of Related Concepts

3.2.1. Turning Points of the Change of Course. The number of path-optimized turning points determines the workload of the controllers and pilots when performing path planning. The number of turning points for redirecting an aircraft in clear weather is usually 2–4, and the maximum number of turns for an aircraft in hazardous weather is 7–8.

3.2.2. Turning Angle of the Aircraft. The turning angle of an aircraft is the amount of constant change in its heading during flight. As shown in Figure 2, θ indicates the angular magnitude of the turn. When radar control is used, the aircraft's turn angle usually does not exceed 60°. The turning angle when flying in restricted areas under hazardous weather conditions usually does not exceed 90°.

3.2.3. Segment Distance. The segment distance is the straight-line distance between two turning points. This distance should be long enough to ensure that an aircraft completes a turn smoothly [28]. Therefore, for a normal flight, the distance between two turning points must exceed 7.4 km, and the turning radius must exceed 3.7 km.

3.3. Modification of Navigation Model and Algorithm. This study focused on the path optimization of a single flight; therefore, in addition to avoiding the flight-restricted area, other factors such as path length, magnitude of turning angle, and number of turns [29] should be considered during the actual modeling process. Under actual circumstances, the turning angle of the flight around the flight-restricted area should not exceed the maximum angle allowed for the aircraft performance, and the turn radius should not be lower than the minimum radius of turn of the aircraft. In the path optimization process, the number of turns should be minimized to avoid aircraft congestion [30]. Therefore, this model adopts the shortest path, minimum turn angle, and minimum number of turns as the objectives, considering the above factors. Simultaneously, to achieve a unified metric for the mathematical modeling of multiple objectives, it was decided to perform dimensionless processing of multiple metrics. The model built based on the above requirements is as follows:

Objective function was as follows:

$$\min C = \omega_1 C_l + \omega_2 C_t + \omega_3 C_a, \tag{7}$$

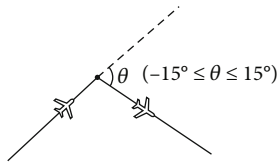

Figure 2: Turning angle of the aircraft.

$$C_l = 1 - \frac{d(p_s,p_D)}{d(p_s,p_l) + \sum_{i=1}^{N_r} d(p_i,p_{i+1}) + d(p_{N_r},p_D)}, \quad (8)$$

$$C_t = \frac{N_t}{N_{\max}}, \quad (9)$$

$$C_a = \frac{\sum_{i=1}^{N_r} \cos\theta}{N_t}. \quad (10)$$

Binding conditions were as follows:

$$\omega_1 + \omega_2 + \omega_3 = 1, \quad (11)$$

$$\omega_1 \geq 0, \omega_2 \geq 0, \omega_3 \geq 0, \quad (12)$$

$$D_{(p_i,p_{i+1})F_i} \geq D_{\min} \forall i = 1 \cdots N_t, j = 1 \cdots n, \quad (13)$$

$$d(p_i,p_{i+1}) \geq d_{\min} \forall i = 1 \cdots N_t, \quad (14)$$

$$\theta < \theta_{\max}, \quad (15)$$

$$N_t \leq N_{\max}. \quad (16)$$

Here are the following definition of parameters:

C_l, C_t, C_a: path length cost, corner number cost, and turning angle change cost

$\omega_1, \omega_2, \omega_3$: path length cost weighting, corner number cost weighting, and turning angle change cost weighting

P_s, P_i, P_D: path starting point, turning point, and path ending point in the path optimization process

$D_{(p_i,p_{i+1})F_i}$: minimum straight-line distance of the path between two turning points from the flight-restricted area

$d(p_s,p_D)$: initial distance between the starting and ending points of the change of course

$d(p_i,p_{i+1})$: straight-line distance between two turning points

$D_{\min}$: minimum distance to deviate from the restricted area, usually 10 km

$d_{\min}$: shortest distance allowed between two turning points, usually 7.4 km

N_t: actual number of corners

$N_{\max}$: maximum number of corners allowed

θ: corner size

$\theta_{\max}$: maximum turning angle allowed

F_i: i-th flight-restricted area

According to the above formula of the objective function, the best path can be obtained when the minimum value of the objective function is considered [31]. Equations (7)–(10) normalize the costs for covering the segment distance, given number of turns, and angle change, and ensure that the smaller the value of C_l, the shorter is the path length after path optimization; the smaller the value of C_t, the smaller is the number of turns, and the smaller the value of C_a, the smaller is the angle change. Equation Equation (11) ensures that the sum of the weight coefficients is 1 and positive; Equation (12) ensures the safety of path optimization, and Equations (13)–(16) limit the segment distance, maximum turning angle, and number of turning points.

4. Model Solution Using Algorithm for Change of Navigation

4.1. RRT Algorithm.* Traditional path planning algorithms such as A^*, shortest path, and genetic algorithms, which require modeling the obstacles in a defined space, are not suitable for solving the path planning of a multidegree-of-freedom aircraft in a complex environment [31]. The fast extended random tree algorithm, that is, the RRT* algorithm, can solve path planning problems in high-dimensional spaces and under complex constraints by performing collision detection on sampled points in the state space. The intention is to generate random points and then search for a step forward from the starting point to the target point. In this manner, it can effectively avoid obstacles, prevent the path from falling into the local minima, and converge faster [32].

According to Figure 3, the steps of the RRT* algorithm are as follows.

Initialize the random tree, adding nodes starting with an arbitrary random tree with only *Q*init.

(1) Execute the *sample* function to obtain a random point *Q*nearest in the map

Iterate through all the nodes in the tree and find the point *Q*nearest with the least cost to reach *Q*rand.

(2) Execute the extend function to obtain the extension point *Q*new of the specified length in the direction of *Q*nearest to *Q* rand and perform collision detection on *Q*new. If false, *Q*nearest is designated as the parent node of *Q*new, and the line connecting the two points is drawn

(3) Determine whether *Q*new has reached the specified target range, if it has, end the loop; otherwise, continue to execute the loop until the target point *Q* rand is found

(4) To increase the practicality of the algorithm, a running time limit and a limit on the number of nodes searched are added to the above process. In other words, the algorithm is judged to have failed after searching for a finite number of nodes within the specified time and not finding the target point

4.2. Improved F-RRT Algorithm.* To solve the problems of strong randomness, poor orientation, slow convergence speed, and poor smoothness of path twisting and tracing

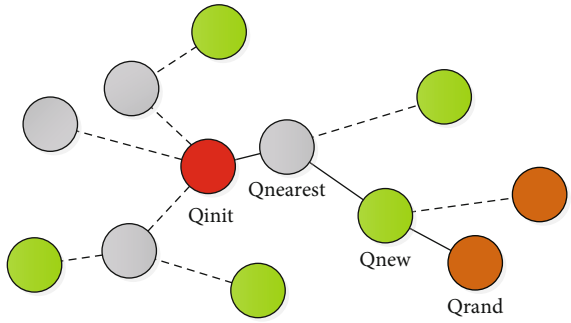

Figure 3: Expansion diagram of the RRT* algorithm.

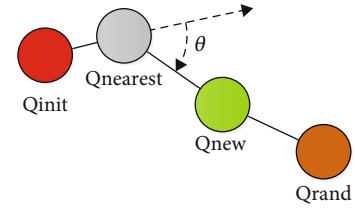

Figure 4: Extended diagram of the F-RRT* algorithm.

of the traditional rapidly exploring random tree* algorithm, an improved RRT* algorithm based on a target-biasing strategy combined with angle restriction is proposed—the F – RRT* algorithm.

First, a target bias threshold is set based on the traditional RRT* algorithm, and the randomness of tree growth is reduced by expanding the internal nodes of the tree through the target bias strategy [33]. Simultaneously, a local expansion mechanism is introduced to avoid the local optimum problem caused by changing the sampling structure. Then, a bidirectional tree growth strategy is used to improve the convergence speed of the algorithm. Finally, redundant nodes are removed, and cubic spline interpolation is used to increase the smoothness of the path.

4.2.1. Extended Step. In the RRT* algorithm, Qinit is extended to Qnearest by a step to obtain the sampling point Qnew, expressed as

$$Qnew = Qinit + \varphi \frac{Qnearest - Qinit}{L(Qinit, Qnearest)}, \tag{17}$$

where φ is the extended step size, and $L(Qinit, Qnearest)$ is the Euclidean distance between the points Qinit and Qnearest.

4.2.2. Target Bias Strategy. The traditional RRT* algorithm has a large range of sampling points and is prone to randomness; therefore, the algorithm search time is longer [34]. By adopting the target bias strategy, the target points are used as sampling points for random tree expansion with a certain probability P, which reduces the number of sampling points and accelerates the efficiency of the random tree expansion [35].

4.2.3. Metric Function. The objective of the metric function is to find the point closest to the selected point Qinit in the randomly expanded tree, namely, Qnearest. In the traditional RRT* algorithm, the metric function of the Euclidean distance is used [36]. The improved F-RRT* algorithm introduces an angle constraint while considering two-point spacing, which makes the entire path smoother.

The distance and angle are unified such that L_d represents the distance function, and M_θ represents the declination function after planning. The node that minimizes the function $M(x_i, x_j)$ is near the node formed in the search extension tree, which can be used as Qnew, as shown in Figure 4.

$$\begin{aligned} L_d &= \frac{d_{\max} - d}{d_{\max}}, \\ M_\theta &= \frac{\theta_{\max} - \theta}{\theta_{\max}}, \\ L(i,j) &= \left\| x_j - x_i \right\|, \\ \theta(i,j) &= \left\| \theta_j - \theta_i \right\|, \\ M(x_i, x_j) &= L_d L(i,j) + M_\theta \theta(i,j), \end{aligned} \tag{18}$$

where the coordinates of the point Qinit are (x_i, y_i), and those of Qnearest are (x_j, y_j); $d_{\max}$ and $\theta_{\max}$ are the maximum distance and maximum angle of the search, respectively.

4.2.4. Path Pruning and Smoothing. The sampling points of the traditional RRT algorithm are highly random, and the generated paths often have numerous redundant nodes [37]. The efficiency in complex environments with many obstacles is low, making path pruning an essential step. The newly generated path nodes are filtered based on the traditional paths to remove redundant nodes, and the pruned paths are then simplified using a greedy algorithm [38].

The specific pruning steps are as follows:

Begin from the starting point Qinit and find all its Q nearest nodes. If the connection between Qinit and one of its Qnearest nodes does not contain obstacles, the Qnearest node is taken as the new node Qnew, and all other Q nearest nodes are deleted. Conversely, if the connection between Qinit and one of its Qnearest nodes encounters obstacles, then the above process is repeated starting from the parent node of the point Qinit.

(1) Find the connection between Qinit and the next Q nearest node until the Q rand node is added to the new path. Finally, the identified nodes are connected by a smooth curve [39]

The K-times B-spline curve equation is as follows:

$$P(u) = \sum_{i=o}^{n} B_{i,k}(u) P_i \, (i = 1, 2 \cdots n), \tag{19}$$

where P_i is the node of the curve control, $B_{i,k}(u)$ is the K-times B-spline basis function, and the node vector $u = [u_0, u_1, \cdots, u_{n+k+1}]$ determines the K-times segmented curve [40].

The basis functions of the triple B-spline curve are as follows:

$$\begin{aligned} b_0 &= \frac{1}{6}\left(-u^3 + 3u^2 - 3u + 1\right), \\ b_1 &= \frac{1}{6}\left(3u^3 - 6u^2 + 4\right), \\ b_2 &= \frac{1}{6}\left(-3u^3 + 6u^2 + 3u + 1\right), \\ b_3 &= \frac{1}{6}u^3. \end{aligned} \tag{20}$$

4.3. Steps to Improve the RRT Algorithm.* The main difference between the improved RRT* algorithm and RRT* algorithm is that the former has two additional recomputation processes for the new node Qnew, which are as follows:

(1) The process of reselecting the parent node for Qnew contains an additional reselection process

(2) An additional rewiring random tree process

Beginning from the starting point Qinit, we search for the "nearest neighbors" in the selected range; that is, we search for the suitable Qnearest node among all the Q nearest nodes from Qinit as an alternative to the parent node Qnew. The distance from the starting point Qinit to the nearest Qnearest node and the distance from the nearest Q nearest node to the parent node Qnew are calculated [41]. The specific steps are illustrated in Figure 5. Figure 5(a) shows the expansion process of the random tree at a certain moment, where node 0 represents the starting point Qinit, node 9 is the newly generated node Qnearest, node 6 is the parent node Qnew of node 9, and the generated path sequence number is 0-4-6-9 with a distance cost of $10 + 5 + 1 = 16$. Figure 5(b) shows the process of identifying the parent node once again based on Figure 5(a). The newly generated Qnearest node, that is, node 9, is considered as the center, and its "near neighbors" are detected within the selected range [42], that is, nodes 5, 6, and 8; they are Q nearest composed of paths 0-1-5-9, 0-4-6-9, and 0-1-5-8-9, and the distances are $3 + 5 + 3 = 11$, $10 + 5 + 1 = 16$, and $3 + 5 + 1 + 3 = 12$, respectively. It can be seen that node 5 is the closest to node 9; therefore, the parent node of node 9 is changed from the original node 6 to node 5, and the results are shown in Figure 5(b).

After reselecting the parent node, the rewiring process is performed on the random tree to further reduce the distance cost of the random extended tree [43]. The specific steps are illustrated in Figure 6. Figure 6(a) shows the rewiring process of the random tree at a certain moment, where node 0 is the starting point Qinit and node 9 is the Qnew node; that is, node 9 is the center, and its "nearest neighbors" are found in the selected range, which are nodes 5, 6, and 8. The distances from the starting point to each "nearest neighbor" node are 8, 15, and 9, respectively. Figure 6(b) shows the rewiring process based on Figure 6(a). Suppose that the parent node of node 5 is not node 1 but node 9, and the path is 0-4-9-5 with distance 17, which is greater than the original path distance 8, then the parent node of node 5 is not changed. Similarly, find the new parent node of node 6. Assuming that it is node 9 with path 0-1-5-9-6 and distance 12, which is less than the original path distance of 15, the parent node of node 6 is changed from node 4 to node 9. Finally, we find the new parent node of node 8. Assuming it is node 9, the path is 0-1-5-9-8, and the distance is 14, which is greater than the original path distance 9; therefore, the parent node of node 8 is not changed [44]. In summary, only the parent node of node 6 must be changed, and the newly generated paths are shown in Figure 6(b). From the above, it can be seen that the significance of rewiring is to reduce the distance cost of some newly generated node paths, thus reducing the distance of the entire optimized path.

The core steps of the improved RRT* algorithm lie in the above two processes, by which the redundant paths can be reduced and optimal paths can be found.

5. Simulation Verification

Southern China is a thunderstorm-prone region. Hence, this study considered the typical route of Guiyang–Guangzhou in China as an example and studied the delineation of the flight-restricted area and optimization of the route affected by thunderstorms and traffic disruption. The Guiyang–Guangzhou route comprises the route points Guiyang–Guilin–Gaoyao–Pingzhou–Guangzhou, as shown in Figure 7.

5.1. Flight-Restricted Area at the Start of the Change of Flight. Through meteorological satellite and Doppler meteorological radar observation images, the Chinese aviation meteorological department observed a band of thunderstorm clouds in the Guiyang–Guilin flight segment and some thunderstorm clouds in the Guilin–Gaoyao flight segment. According to the method introduced in Section 1, the initial flight-restricted area is set up; the area has a radar echo $\geq$ 41dBZ at a certain time t, as shown in Figure 8 [45]. To facilitate the analysis, the flight restricted area obtained following the processing in Figure 8 is analyzed separately in Figure 9. To simplify the graphical representation, Guiyang is denoted as A, Guilin as B, Gaoyao as C, Pingzhou as D, and Guangzhou as E in Figure 9. The boundary points are $s_1 - s_{10}$, with the location of Guiyang Airport in China as the origin, magnetic north as the positive direction of the y-axis, and magnetic west as the positive direction of the x-axis [46]. The coordinates of Guiyang, Guilin, and Gaoyao in this map are (0, 0), (152.50, −53.26), and (394.30, −256.72), respectively. The coordinates of S1–S10 are (213.64, −24.78), (265.64, −153.32), (299.64, −97.82), (304.14, −219.32), (249.64, −182.82), (202.14, −187.20), (174.14, −5.32), (174.14, −32.82), (97.26, −34.14), and (97.26, −6.33), respectively. The lumpy thunderstorm

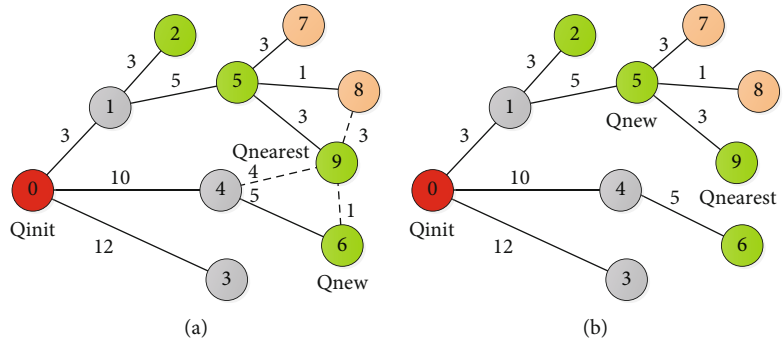

Figure 5: Process of reselecting the parent node.

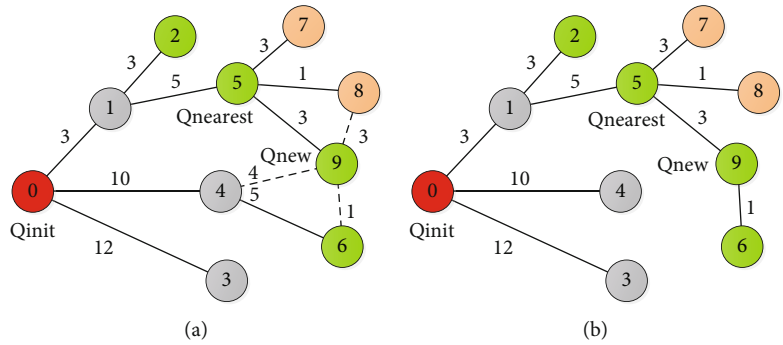

Figure 6: Process of rewiring a random tree.

Figure 7: High-altitude aerial map of Guiyang–Guangzhou route in China.

cloud AFFA was observed to be moving in the direction of 40° east of north; the rates of movement were $v_1 = 40$ km/h and $v_2 = 10$ km/h, and the safety margin was $L_1 = 25$ km; the expansion speed of the thunderstorm boundary point was $V_{s2} = 20$ km/h [47]. The direction of movement of the banded thunderstorm cloud BFFA was 20° south by west; the rates of movement were $v_3 = 30$ km/h and $v_4 = 10$ km/h, and the safety margin was $L_2 = 25$ km; the expansion speed of the thunderstorm boundary point was $V_{s1} = 20$ km/h, and the acceleration was $a = 5$ km/h2. The flight from Guiyang to Guangzhou was expected to start at a certain time t [48], $t_0 = 0.5$ h after arriving in Guilin. The cruising speed was $v_s = 800$ km/h.

Based on the initial flight-restricted area, the dynamic flight-restricted areas AFFA2 and BFFA2 are determined according to the prediction method introduced in Sections 2 and 2.1, as shown in Figure 10. Owing to the uncertainty of the thunderstorm movement, the boundary of the initial flight restricted area is shifted outward by 25 km [49]. Then, it is expanded outward with a speed $v = 20$ km/h and an acceleration $a = 3$ km/h using the uniform speed prediction method to obtain the finalized dynamic flight-restricted areas AFFA2 and BFFA2. First, the movement of the flight-restricted area was considered according to the boundary point *Si* of the affected area by using the Graham algorithm. The initial dynamic flight-restricted areas AFFA1 and BFFA1 were determined after an elapsed time $t = 0.5$ h, as shown in Figure 10. Because of the uncertainty of the thunderstorm movement, the flight-restricted area boundary was shifted outward by 25 km [49]. Considering the variation in the thunderstorm boundary, the finalized dynamic flight-restricted areas AFFA2 and BFFA2 were obtained by expanding outward with velocity $v = 20$ km/h and acceleration $a = 3$ km/h2 [50]. The coordinates of the vertices are (210.44, −4.70), (255.64, −53.32), (319.64, −96.82), (314.14, −239.32), (233.64, −221.82), (197.14, −197.20), (144.14, 4.33), (144.14, −42.82), (76.26, −42.82), and (76.26, 4.33) assuming that the actual flight-restricted areas are AFFA2 and BFFA2, as shown in Figure 11, and path planning is performed based on them.

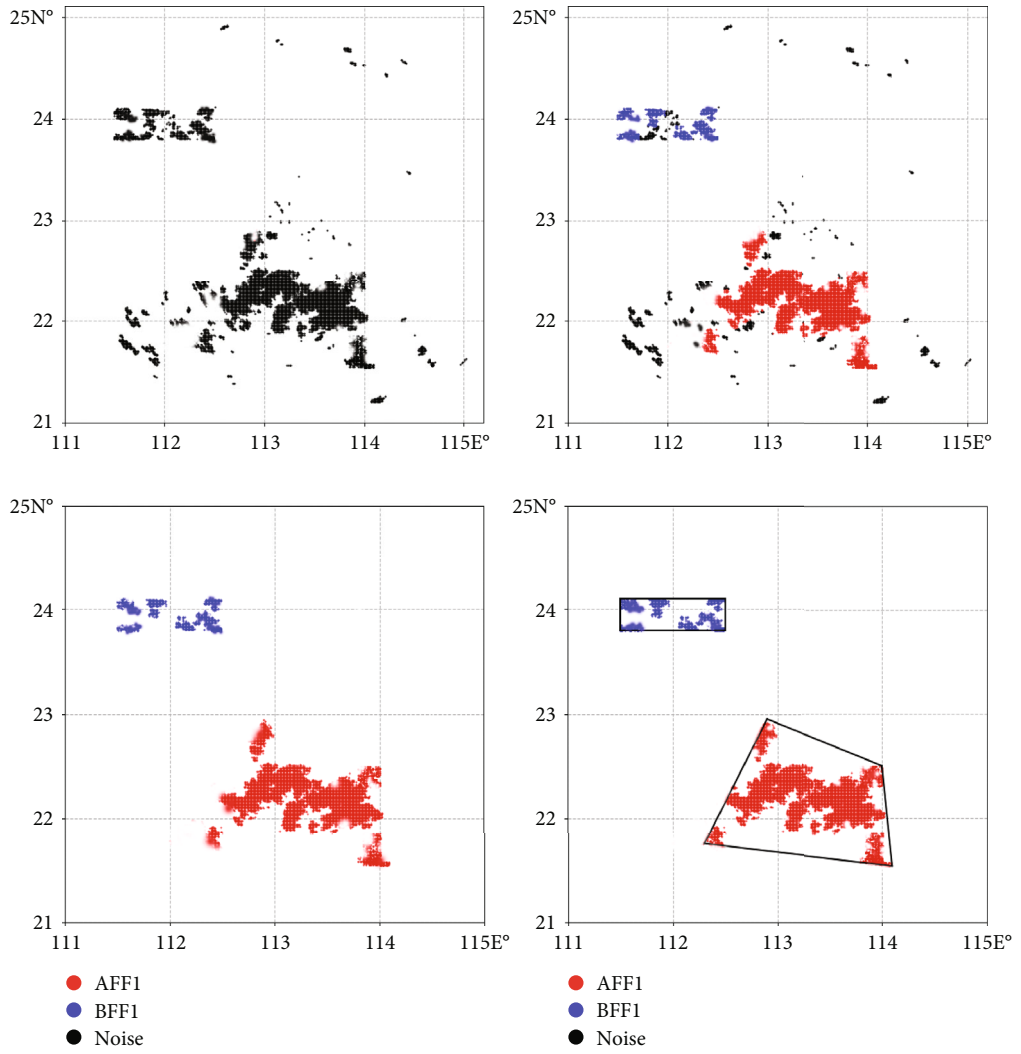

Figure 8: The initial flight-restricted areas.

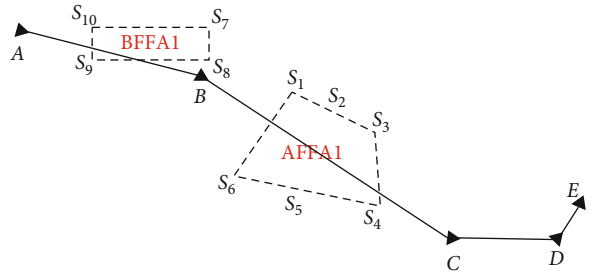

Figure 9: Initial boundary of hazardous weather on the Guiyang–Guangzhou route in China.

The above figure depicts the delineation of the flight-restricted area. The next step is to set the model parameters and solve the model. First, we set up the simulation environment with the location of Guiyang Airport in China as the origin, magnetic south as the positive y-axis direction, magnetic west as the positive x-axis direction, starting point as [0,0], ending point as [860,480], and extended step as 20 m; the standard deviation $\sigma = 5.6$, $k_p = 420000$. The standard threshold of the path length was 586 m. The search was terminated when the number of sampled nodes was greater than 18000 or the path length was less than the standard threshold.

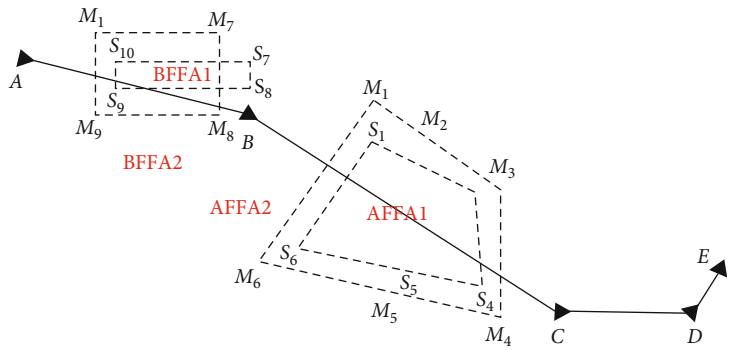

Figure 10: Dynamic flight-restricted area on the Guiyang–Guangzhou route in China.

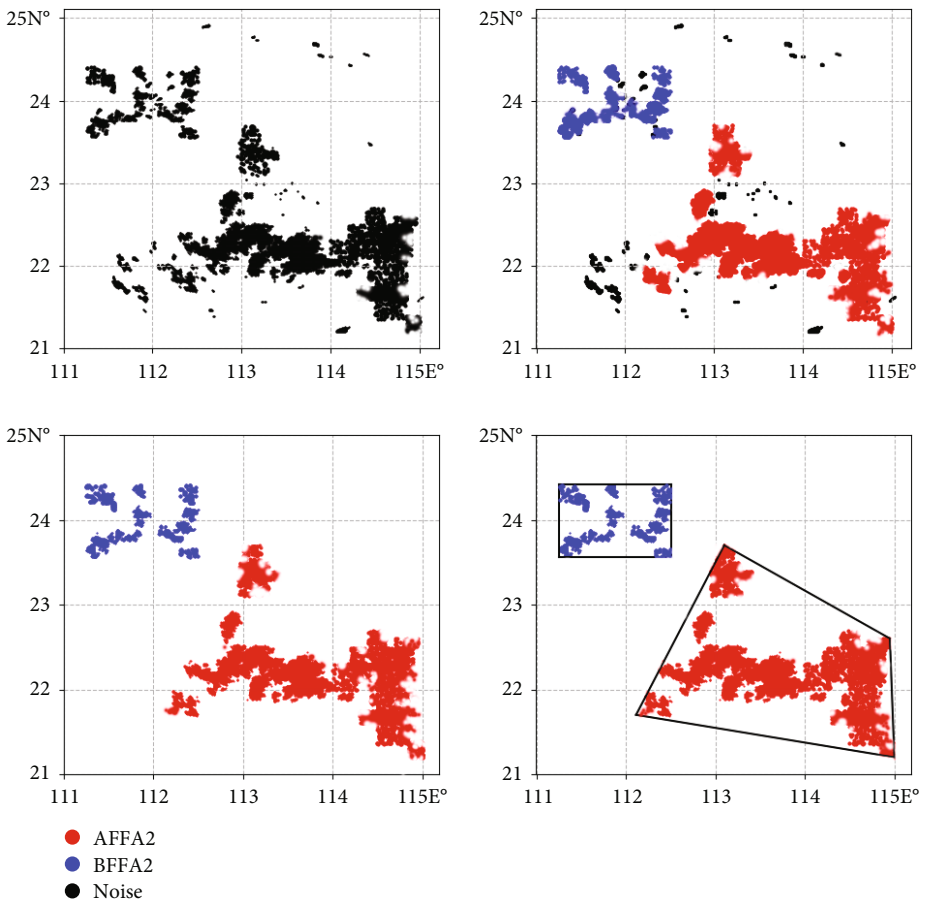

Figure 11: The finalized dynamic flight-restricted areas.

The results obtained using the improved F-RRT* algorithm based on the above parameters are shown in Figure 12. From the following graphs, it can be seen that this algorithm enables the flight to bypass the flight-restricted area affected by dangerous weather more smoothly and finds the least costly path more quickly when compared with the other algorithms [51].

The paths obtained by the following four algorithms were compared and analyzed in terms of the path planning time, number of turning points, and path distance. The analysis results are shown in Figure 13. In terms of the path planning time, the F-RRT* algorithm uses the shortest time to identify a path. The effect is significant as the number of sampling points increases. In terms of the number of turning points, it can identify more valid nodes, whereas in terms of the path distance, it also determines the shortest path. The specific parameters are shown in Table 1. The simulation results show that the F – RRT* algorithm has a shorter path

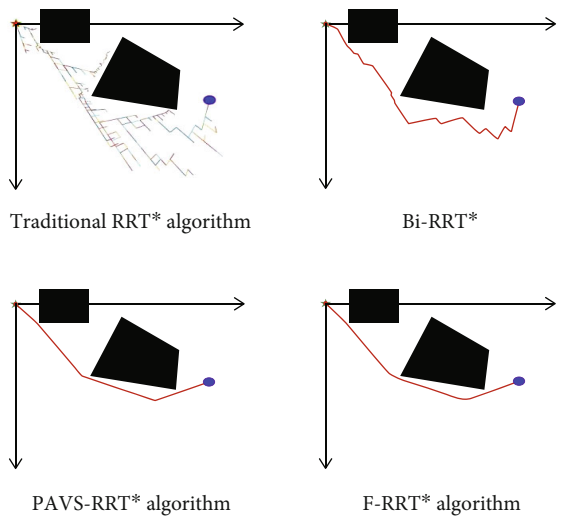

Figure 12: Path planning effect.

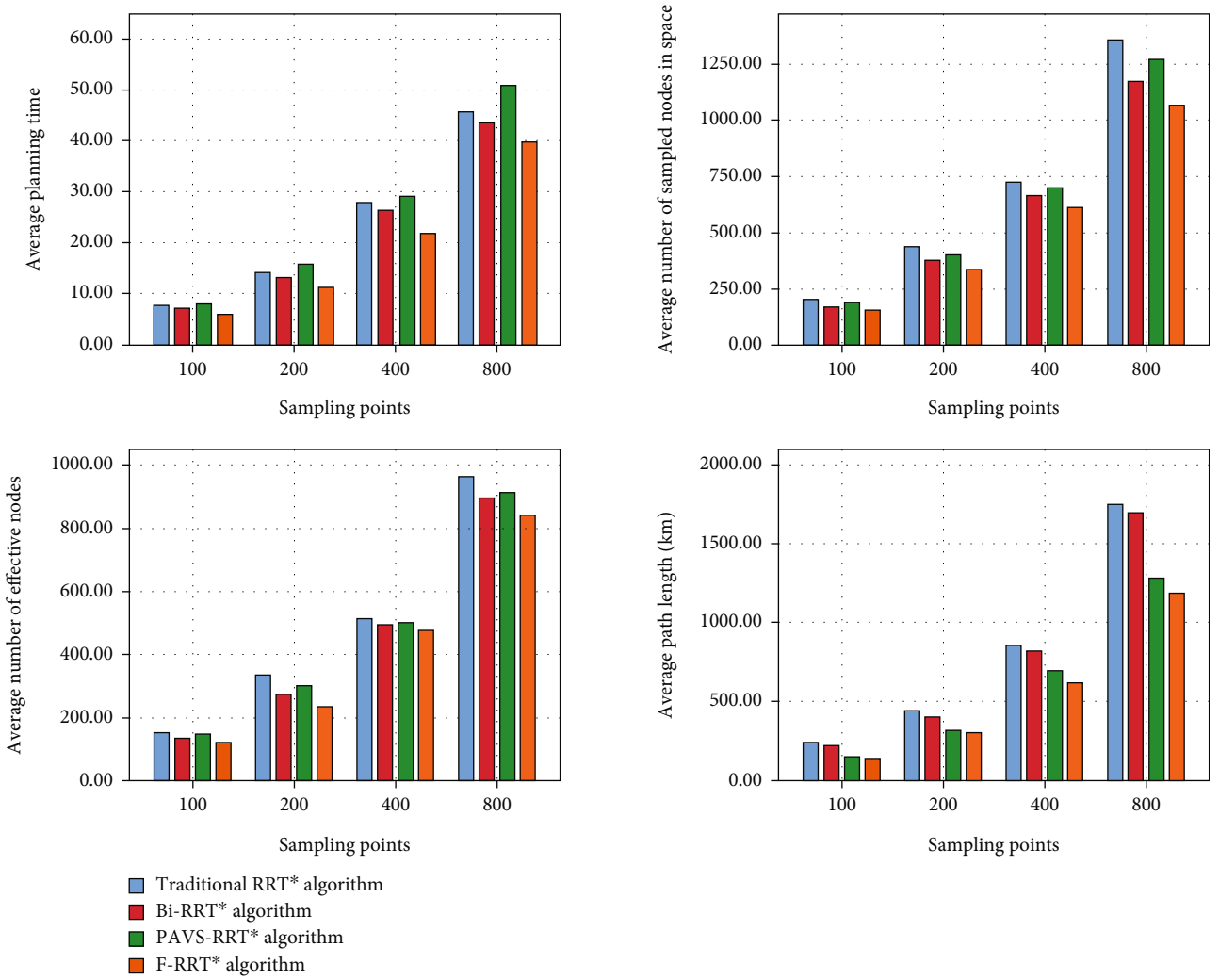

Figure 13: Comparison and analysis of the capability of four algorithms.

Table 1: Comparative performance analysis of various RRT* algorithms.

Algorithm	Average planning time (s)	Average number of sampled nodes in space	Average number of effective nodes	Average path length (km)
Traditional RRT* algorithm	45.66	1355	964	1847.49
Bi-RRT* algorithm	43.56	1168	897	1693.65
PAVS-RRT* algorithm	50.69	1267	912	1280.77
F-RRT* algorithm	39.68	1065	843	1183.84

planning time, fewer turning points, and shorter and smoother paths than the other *RRT** algorithms. It can be better used for flight path planning.

6. Conclusion

This paper presented an in-depth study on the redirecting strategy of flight paths under hazardous weather conditions. An improved F-RRT* algorithm based on the target-biasing strategy combined with angle restriction was proposed. The traditional RRT* algorithm has been conventionally used for robot path planning; it has never been used for flight-path planning. In this study, we improved the traditional RRT* algorithm based on the characteristics of flight and derived an algorithm applicable to flight-path optimization, that is, the F-RRT* algorithm.

The main aspects of this study and its findings are as follows.

(1) The radar data for the Guiyang–Guangzhou route in China were obtained, processed and filtered to determine the initial hazardous weather area on the route based on the radar echo values. Then, the Graham algorithm was proposed to predict the flight-restricted area according to the boundary change of hazardous weather to obtain the final flight-restricted area on the route. The flight-restricted area was divided into three categories based on its shape and distribution characteristics: block-shaped, strip-shaped, and scattered. The concepts related to changes in flight were briefly explained

(2) Based on the above analysis, the model was constructed as a grid environment model with the flight route distance, route turn size, and number of turning points as the objectives, and the model was explained

(3) An improved F-RRT* algorithm based on a target-biasing strategy combined with angle restriction was proposed based on the flight characteristics. It solved the problems of traditional path planning algorithms, such as strong randomness, poor guidance, slow convergence speed, poor path curvature, and tracing smoothness

(4) The model obtained under this algorithm was simulated and analyzed for the Guiyang–Guangzhou route in China as an example. The obtained rerouting paths were compared in terms of the path planning time, path distance, and number of turning points. The simulation results showed that the F-RRT* algorithm had shorter planning time, fewer turning points, smaller turning angle, and smoother path than the other RRT* algorithms. This method is effective in reducing flight delays while ensuring safety interval standards

Although the achievements of this study are significant, several aspects can be further improved in future research, for example, the establishment of a more detailed dynamic environment model, improvement of the metric for bad weather, and consideration of three-dimensional modeling of the flight paths. It is also possible to conduct in-depth research on multiflight and multiroute flight plans and expand from diversion path planning for one route to that for route networks.

Data Availability

The data containing the results of this study are related to flight safety of the aircraft and are copyrighted. Upon request, part of the data can be obtained from the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the China Meteorological Administration (grant number 0052024), Research Start-Up Project of the Civil Aviation Flight University of China (grant number 09005001), Special Project of the Civil Aviation Flight University of China (TS-06), Postgraduate Research and Innovation Fund Project of the Civil Aviation Flight University of China (XSY2022-37), and Traffic Engineering Advantages and Characteristic Discipline Construction Project of the Civil Aviation Flight University of China (D202103). We are grateful for the support provided by the funding agencies. We would like to thank Editage (http://www.editage.cn/) for the English language editing.

References

[1] Y. K. Zheng, Y. Wang, Y. Sun, Y. Qi, and N. Wang, "Multi-aircraft rerouting method under dangerous weather," in *Asia Conference on Mechanical Engineering and Aerospace*

Engineering (MEAE), E. D. P. Sciences, Ed., vol. 198, China Univ Geosciences, Wuhan, PEOPLES R CHINA, 2018.

[2] B. D. Dancila and R. M. Botez, "Vertical flight path segments sets for aircraft flight plan prediction and optimisation," *Aeronautical Journal*, vol. 122, no. 1255, pp. 1371–1424, 2018.

[3] U. Ahlstrom and E. Jaggard, "Automatic identification of risky weather objects in line of flight (AIRWOLF)," *Transportation Research Part C-Emerging Technologies*, vol. 18, no. 2, pp. 187–192, 2010.

[4] Y. X. Liu, C. Y. Han, H. L. Qi, and Z. L. Zhu, "Aircraft rerouting decision-making model under severe weather," in *3rd International Conference on Information Science and Control Engineering (ICISCE)*, pp. 814–818, IEEE, Beijing, PEOPLES R CHINA, 2016.

[5] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous UAVs," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[6] W. C. Ding and D. Sui, "Pre-flight rerouting combining a∗ algorithm and AHP under severe weather," *Journal of Physics: Conference Series*, vol. 1187, no. 4, 2019.

[7] N. Chao, Y. K. Liu, H. Xia, M. J. Peng, and A. Ayodeji, "DL-RRT∗ algorithm for least dose path re-planning in dynamic radioactive environments," *Nuclear Engineering and Technology*, vol. 51, no. 3, pp. 825–836, 2019.

[8] B. Song, G. Qi, and L. Xu, "A Survey of three-dimensional flight path planning for unmanned aerial vehicle," in *31st Chinese Control And Decision Conference (CCDC)*, pp. 5010–5015, IEEE, Nanchang, PEOPLES R CHINA, 2019.

[9] J. C. Chen, F. Y. Ling, Y. Zhang, T. You, Y. F. Liu, and X. Y. Du, "Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system," *Swarm and Evolutionary Computation*, vol. 69, article 101005, 2022.

[10] J. C. Chen, Y. Zhang, L. W. Wu, T. You, and X. Ning, "An adaptive clustering-based algorithm for automatic path planning of heterogeneous UAVs," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[11] V. Singh and K. E. Willcox, "Methodology for path planning with dynamic data-driven flight capability estimation," *AIAA Journal*, vol. 55, no. 8, pp. 2727–2738, 2017.

[12] Z. A. Ali, B. Angulo, V. Golovin, and K. Yakovlev, "Empirical evaluation of theta-RRT and GRIPS algorithms," in *2021 International Siberian Conference on Control and Communications (SIBCON)*, pp. 1–6, Kazan, Russia, 2021.

[13] M. L. Ding, "Optimal path algorithm analysis of GPS navigation based on A∗ algorithm," in *3rd International Conference on Green Power, Materials and Manufacturing Technology and Applications*, pp. 982–985, Trans Tech Publications Ltd, Wuhan, PEOPLES R CHINA, 2013.

[14] Z. Laboudi and S. Chikhi, "Comparison of genetic algorithm and quantum genetic algorithm," *International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 243–249, 2012.

[15] Y. J. Lin and X. J. Chen, "Simulated annealing algorithm improved BP learning algorithm," *Applied Mechanics and Materials*, vol. 513, pp. 734–737, 2014.

[16] G. Ping, X. Chunbo, C. Yi, L. Jing, and L. Yanqing, "Adaptive ant colony optimization algorithm," in *International Conference on Mechatronics and Control (ICMC)*, pp. 95–98, IEEE, Jinzhou, PEOPLES R CHINA, 2014.

[17] L. Jia, Y. Huang, T. Chen, Y. Guo, Y. Yin, and J. Chen, "MDA + RRT: a general approach for resolving the problem of angle constraint for hyper-redundant manipulator," *Expert Systems with Applications*, vol. 193, article 116379, 2022.

[18] E. Andrés, D. González-Arribas, M. Soler, M. Kamgarpour, and M. Sanjurjo-Rivo, "Informed scenario-based RRT* for aircraft trajectory planning under ensemble forecasting of thunderstorms," *Transportation Research Part C: Emerging Technologies*, vol. 129, article 103232, 2021.

[19] M. Kang, Y. Zeng, H. Yao, J. Zhang, and Y. Liu, "RRT algorithm under holonomic conditions for UAVs," in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 716–720, Fukuoka, Japan, 2019.

[20] C. Zong, X. Han, D. Zhang, Y. Liu, W. Zhao, and M. Sun, "Research on local path planning based on improved RRT algorithm," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 8, pp. 2086–2100, 2021.

[21] H. You, X. Jianjuan, and G. Xin, *Radar Data Processing with Applications*, John Wiley & Sons, 2016.

[22] P. T. An, P. T. T. Huyen, and N. T. Le, "A modified Graham's convex hull algorithm for finding the connected orthogonal convex hull of a finite planar point set," *Applied Mathematics and Computation*, vol. 397, article 125889, 2021.

[23] J. Bruce, "The Development of International Civil Aviation Regulations," *The Lawbook Exchange, Ltd*, pp. 1899–1944, 2021.

[24] M. Gatto and P. Widmayer, "On the robustness of graham's algorithm for online scheduling," in *Workshop on Algorithms and Data Structures*, pp. 349–361, Springer, 2007.

[25] C. Schwarz and K.-U. Hahn, *Simplified Hazard Area Prediction Method-SHAPe*, DLR-Interner Bericht, 2011.

[26] J. A. Goldman, "Path planning problems and solutions," in *Proceedings of National Aerospace and Electronics Conference (NAECON'94)*, pp. 105–108, IEEE, Dayton, OH, USA, 1994.

[27] S. Li, C. Ma, Q. Li, J. Zeng, and L. Wang, "Application of improved ant colony algorithm in flight path planning," in *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, pp. 763–771, IEEE, Chongqing, China, 2020.

[28] J. S. Holub, *Improving particle swarm optimization path planning through inclusion of flight mechanics, [Ph.D. thesis]*, Iowa State University, 2010.

[29] F. Wang, "A new algorithm of flight rerouting path planning in severe weather," *Applied Mechanics and Materials*, vol. 182, pp. 1895–1899, 2012.

[30] W. L. Liu, Y. J. Gong, W. N. Chen, Z. Q. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5094–5109, 2020.

[31] S. C. Zhou, L. N. Xing, X. Zheng, N. Du, L. Wang, and Q. F. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1430–1442, 2021.

[32] L. He, A. Zhao, X. Wang, Z. Zhang, P. Wang, and R. Wu, "Path planning method for general aviation under hazardous weather using heuristic algorithm," in *2019 5th International*

Conference on Transportation Information and Safety (ICTIS), pp. 920–926, IEEE, Liverpool, UK, 2019.

[33] K. Naderi, J. Rajamäki, and P. Hämäläinen, "RT-RRT∗ a real-time path planning algorithm based on RRT," in *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pp. 113–118, New York, 2015.

[34] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, pp. 1178–1183, IEEE, Las Vegas, NV, USA, 2003.

[35] K. Wei and B. Ren, "A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm," *Sensors*, vol. 18, no. 2, p. 571, 2018.

[36] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT∗: An improved path planning algorithm for mobile robots," *Expert Systems with Applications*, vol. 152, article 113425, 2020.

[37] M. Svenstrup, T. Bak, and H. J. Andersen, "Minimising computational complexity of the rrt algorithm a practical approach," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5602–5607, IEEE, Shanghai, China, 2011.

[38] J.-Z. Song, B. Dai, E.-Z. Shan, and H.-G. He, "An improved RRT path planning algorithm," *Acta Electronica Sinica*, vol. 38, no. 2A, pp. 225–228, 2010.

[39] F. Q. Zhao, X. He, and L. Wang, "A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem," *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5291–5303, 2021.

[40] O. Adiyatov and H. A. Varol, "A novel RRT∗-based algorithm for motion planning in dynamic environments," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1416–1421, IEEE, Takamatsu, Japan, 2017.

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

[42] B. Kim, T. T. Um, C. Suh, and F. C. Park, "Tangent bundle RRT: a randomized algorithm for constrained motion planning," *Robotica*, vol. 34, no. 1, pp. 202–225, 2016.

[43] F. Q. Zhao, L. X. Zhang, J. Cao, and J. X. Tang, "A cooperative water wave optimization algorithm with reinforcement learning for the distributed assembly no-idle flowshop scheduling problem," *Computers & Industrial Engineering*, vol. 153, article 107082, 2021.

[44] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT∗: Optimal sampling-based motion planning with automatically derived extension heuristics," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2537–2542, IEEE, Saint Paul, MN, USA, 2012.

[45] C. Tovar, D. Aranguren, J. López, J. Inampués, and H. Torres, "Lightning risk assessment and thunderstorm warning systems," in *2014 International conference on lightning protection (ICLP)*, pp. 1870–1874, IEEE, Shanghai, China, 2014.

[46] W. Yingxun and C. Zongji, "Genetic algorithms (GA) based flight path planning with constraints," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 25, no. 13, 1999.

[47] M.-C. Mah, H.-S. Lim, and A. W.-C. Tan, "UAV relay flight path planning in the presence of jamming signal," *IEEE Access*, vol. 7, pp. 40913–40924, 2019.

[48] X. Fu and X. Gao, "Genetic algorithm with adaptive immigrants for dynamic flight path planning," in *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*, pp. 630–634, IEEE, Xiamen, China, 2010.

[49] D. Šišlák, P. Volf, and M. Pechoucek, "Flight Trajectory Path Planning," in *Proceedings of the 19th International Conference on Automated Planning & Scheduling (ICAPS)*, pp. 76–83, Menlo Park, 2009.

[50] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous UAV path planning and estimation," *IEEE Robotics & Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.

[51] W. Yanyang, W. Tietao, and Q. Xiangju, "Study of multi-objective *fuzzy* optimization for path planning," *Chinese Journal of Aeronautics*, vol. 25, no. 1, article S1000936111603610, pp. 51–56, 2012.