Research Article

UAV Path Planning Based on Improved A∗ and DWA Algorithms

Xiong Bai,[1] **Haikun Jiang,**[2] **Junjie Cui,**[1] **Kuan Lu** ⓘ**,**[1] **Pengyun Chen** ⓘ**,**[1] **and Ming Zhang**[3]

[1]*School of Mechatronic Engineering, North University of China, Taiyuan 030051, China*
[2]*Shandong Product Quality Inspection Research Institute, Jinan 250000, China*
[3]*School of Energy and Power Engineering, North University of China, Taiyuan 030051, China*

Correspondence should be addressed to Kuan Lu; elukuan@163.com and Pengyun Chen; chenpengyun@nuc.edu.cn

This work proposes a path planning algorithm based on A∗ and DWA to achieve global path optimization while satisfying security and speed requirements for unmanned aerial vehicles (UAV). The algorithm first preprocesses the map for irregular obstacles encountered by a UAV in flight, including grid preprocessing for arc-shaped obstacles and convex preprocessing for concave obstacles. Further, the standard A∗ algorithm is improved based on UAV's flight environment information and motion constraints. Further, the DWA algorithm's limitations regarding local optimization and long planning time are mitigated by adaptively adjusting the evaluation function according to the UAV's safety threshold, obstacles, and environment information. As a result, the global optimal path evaluation subfunction is constructed. Finally, the key points of the global path are selected as the subtarget points of the local path planning. Under the premise of the optimal path, the UAV real-time path's efficiency and safety are effectively improved. The experimental results demonstrate that the path planning based on improved A∗ and DWA algorithms shortens the path length, reduces the planning time, improves the UAV path smoothness, and enhances the safety of UAV path obstacle avoidance.

1. Introduction

Unmanned aerial vehicles (UAVs) are widely used in industry, life sciences, logistics, and other fields. In recent years, path planning has emerged as a critical problem for UAVs. UAV path planning sets the start and end points in the UAV environment map and plans a collision-free, optimal, and safe path using a relevant path planning algorithm.

At the core of the UAV, path planning is the path planning algorithm. Based on the employed planning scenarios, path planning can be divided into global path planning and local path planning. Global path planning utilizes the environment map with known obstacles, while local path planning relies on the environment map with unknown obstacles. The main algorithms used in global path planning are, for example, A∗ [1, 2], Dijkstra [3], and RPT [4]. The local path planning algorithms include dynamic window algorithms (DWA) [5, 6], artificial potential field method (APF) [7, 8], and ant colony algorithm (ACO) [9]. Many scholars researching the UAV path planning assume that

the obstacles in the UAV environment are regular in shape (e.g., circle or rectangle). However, disregarding the obstacles' irregularities prevents accurate representation and planning of the UAV flight path.

Lyu et al. proposed a path planning algorithm based on graph search. The improved algorithm uses the Floyd algorithm to weight different paths, which greatly reduces the number of path points and effectively improves the smoothness of the path [10]. Turker et al. study the problem of UAV radar evasion by regarding a radar as a circular no-fly zone and realizing the path planning on a two-dimensional plane via the simulated annealing algorithm [11]. Yao and Wang propose an adaptive ant lion optimization algorithm for path planning [12]. The authors verified the proposed algorithm's effectiveness but only considering circular and rectangular no-fly zones. Thus, it is more suitable for maps with ideal obstacle boundaries [12]. Similarly, Xu et al. model obstacles as circular no-fly zones and optimize the planning path based on the Laguerre diagram [13]. Boulares and Barnawe et al. proposed a regression

algorithm, which used the predicted trajectory of the missing target to plan the path of the UAV search task [14]. Belkadi et al. proposed a distributed UAV trajectory generation strategy, which solved some problems in a group of UAV control and can be applied to large-scale search, monitoring, inspection, and rescue [15]. Huang et al. proposed a coordinated path planning method for multiple UAVs based on k-order smoothing and constructed a complex environment composed of multiple threat sources. Using the improved ant colony algorithm, a k-order smoothing algorithm is proposed to obtain a more flexible path [16].

DWA is a commonly used local path planning algorithm, which has good obstacle avoidance ability. However, it is prone to falling into a local optimum and not reaching the specified target point. Moreover, the increase in complexity of the UAV application scene and environment may render the standard DWA incapable of solving the complex path planning problem. Consequently, DWA typically needs to be improved through combination with global path planning algorithms.

According to Yulong et al., in solving the UAV path planning problem, the decoupling strategy is introduced, and the messenger UAV path planning problem is described as a dynamic traveling salesman problem with neighborhood [17]. Huo et al. proposed a hybrid differential symbiotic organisms search algorithm (HDSOS), which combines the differential evolution mutation strategy with the improved symbiotic organisms search (SOS) strategy [18]. The algorithm maintains the systems' local search capability while simultaneously achieving a good global search performance [18]. To solve the problem of UAV route planning under different threats in a complex environment, Phung and Ha proposed a particle swarm optimization algorithm that not only is superior to the classical particle swarm optimization, phase swarm optimization, and quantum swarm optimization but also outperforms the genetic algorithm [19]. Jeauneau et al. proposed two methods for UAV path planning in a real 3D environment. A single path is provided based on the A-star method, and multiple paths are provided by the genetic algorithm using Pareto Frontier (PF). The generated paths must meet the dynamic characteristics of vehicles [20].

When establishing the environment, the studied literature regards the planning space as too perfect, rendering it impractical. Such path planning algorithms are suitable for simple environmental maps. Considering that most obstacles in the real environment are irregular in shape and UAVs are often regarded as particles, the direct application of the listed methods requires expanding the range of obstacles and affects the planning results. Thus, an improvement is required.

To support path planning in complex environments, this paper first studies the irregular obstacles that a UAV may encounter in a flight. Such a process includes grid processing of arc obstacles and convex processing of concave obstacles. Next, the cost function is improved by using the obstacle weight coefficient in the environment map. The standard A∗ algorithm is advanced by incorporating the UAV motion constraints and a safety threshold. While considering the incompatibility of speed and safety in DWA, the algorithm is improved through an adaptive speed evaluation function that considers the density of obstacles and the safety threshold. Finally, critical nodes in the global path planning are taken as local subobjective points to improve the algorithm while relying on the improved A∗ and DWA algorithms. As a result, the optimal, shortest path with greater smoothness from the start to the end is planned.

2. Environment Modeling and Obstacle Pretreatment

Most of the obstacles in the flight environment of UAV are irregular in shape, and UAV is often regarded as a particle. The UAV flight environment is directly regarded as a regular map, which is easy to expand the range of obstacles and affect the planning results, so the environment map is first processed.

2.1. Grid Method for Airspace Modeling. Assuming that the UAV's altitude is constant during flight, the UAV motion in three-dimensional space is simplified to two-dimensional motion. Suppose that the following UAV path-planning conditions are known:

(1) Starting point and target point

(2) UAV's physical performance limitations

(3) Information regarding irregular obstacles

Further, note that a cruising UAV can be regarded as a particle.

2.2. Gridding Treatment of Arc Obstacles. This paper utilizes the idea of a rough set to obtain a polygon approximating the arc-shaped obstacles [21]. The main method first determines a grid of the arc obstacle's outer edge line, along with its positioning grid. Then, the polygon processing is performed, and each arc grid's positioning points are connected, as shown in Figure 1(b). The specific steps are as follows:

(Step 1) Determine the outer edge line (L) of the arc obstacle (B). Further, determine the grid position of each arc line L_i, and denote the marked grid elements as M_i. Regard the centers of the marked grid elements as the locating points, and denote them as $P_i, i = 1, 2, \cdots, n$, where n is the number of segments of the outer edge line

(Step 2) Connect the locating points $P_1, P_2, \cdots, P_n, P_1$, obtaining the polygon $P_1 P_2 \cdots P_n P_1$ by regarding each $M_1, M_2, \cdots, M_n, M_1$ clockwise from the starting marked grid element. Consequently, obtain the result of obstacle B polygon processing, $B \approx P_1, P_2 \cdots P_n P_1$

2.3. Convex Filling of Concave Polygon Obstacles. The convex filling process commences by determining the coordinates of each polygon vertex in the grid graph. Starting from the first
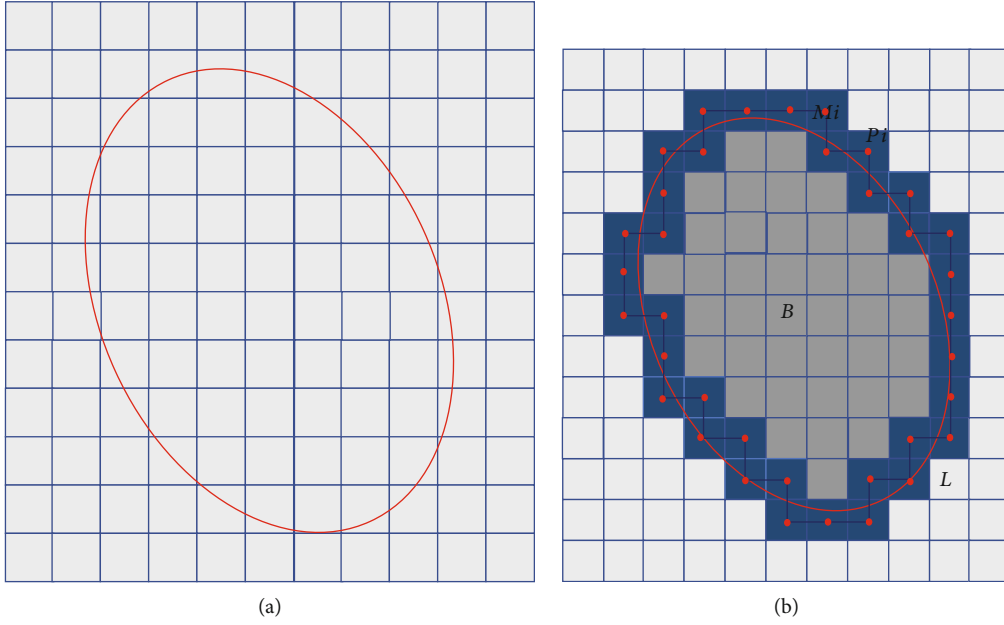
FIGURE 1: Comparison of the gridding treatment for arc-shaped obstacles. (a) Before treatment. (b) After treatment.

grid element, the polygon vertices are denoted $S_1, S_2, \cdots, S_m$, where m is the number of vertices of the polygon. If the inner angle formed by the two adjacent edges is greater than 180 degrees, the corresponding vertex S_i is a concave point. Otherwise, the vertex is a convex point. In the path-searching algorithm, if the path point falls into the concave area, the next path point must be placed outside the concave area to complete the flight mission. Concave areas affect the UAV path quality and increase the number of invalid path points, affecting the solution speed.

Figure 2 shows an example of the convex filling process. The steps are as follows.

(Step 1) Determine the concave polygon's vertices S_1, $S_2, \cdots, S_7$, and judge the concavity and convexity of each vertex according to the concavity and convexity rules (Figure 2(a))

(Step 2) Based on Step 1, start from the starting convex point S_1, and connect other convex points clockwise, yielding the polygon $S_1 S_3 S_4 S_5 S_6 S_1$ as the convex result (i.e., $F \approx S_1 S_3 S_4 S_5 S_6 S_1$). Then, the obstacles are gridded according to the arc obstacles, as shown in Figure 2(b)

3. Improved A∗ Algorithm

There are many turning points and large angles in the path planned via the standard A∗, which is not conducive to a UAV flight. Thus, this work improves the standard A∗ according to the motion characteristics and environmental information of UAVs in flight.

3.1. Improvement of the Heuristic Function. The evaluation function of A∗ is composed of a cost function $g(n)$ and a heuristic function $h(n)$, and the algorithm's optimal search-

ing performance depends on the selection of the heuristic function. The improved A∗ algorithm introduces the obstacle weight coefficient into the heuristic function, as shown in Equation (2). The obstacle weight coefficient expresses the grid map complexity, and the environmental information is analyzed.

The obstacle weight coefficient is defined as the proportion of the number of obstacles in the current map and the number of grid cells in the whole grid map. Let N stand for the number of obstacle grid cells and denote UAV starting and ending coordinates as (x_s, y_s) and (x_g, y_g), respectively. The obstacle weight coefficient is defined as

$$K = \frac{N}{\left(|x_s - x_g|\right) \times \left(|y_s - y_g|\right)} \quad (K \in (0, 1)), \tag{1}$$

$$f(n) = a * g(n) + \left(\frac{\pi}{2} - \arctan K\right) * h(n), \tag{2}$$

where $g(n)$ is the cost from the starting node to the current node, $h(n)$ is the heuristic function value from the current node to the target node, (x_n, y_n) are the current node's coordinates, and a is the weight of the cost function $g(n)$. The coefficient a is calculated as the ratio of the distance from the current node to the target point and the distance from the starting node to the target point.

According to equation (2), the improved algorithm sets the weight of the adaptive heuristic function. When the obstacle weight coefficient K is small, the weight of the heuristic function is increased. The A∗ algorithm reduces the search space, improves the speed of path planning, and effectively reduces the inflection points and turning points of the path. When the obstacle weight coefficient is large, reduce the weight of the heuristic function and increase the
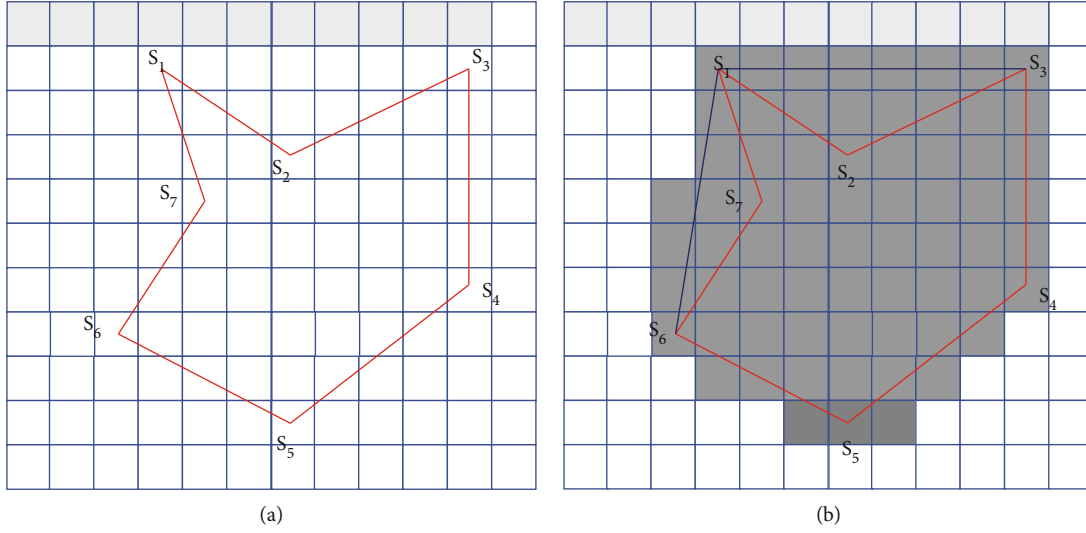
FIGURE 2: The convex filling of a concave polygon. (a) Before treatment. (b) After treatment.

search space to avoid the algorithm falling into local optimization.

3.2. Determining the UAV Safety Distance. The improved A∗ sets the safety distance between the path node and the obstacle to prevent the UAV from colliding with the obstacle. As shown in Figure 3, the vertical distance OE (the vertical distance from point O to line KG) from the obstacle to the path is compared with the preset safety distance to judge whether the planned path is safe and feasible.

Suppose the coordinates of node K are (x_1, y_1), and the coordinates of target node G are (x_2, y_2). Denote the coordinates of obstacle node O as (x_3, y_3) and the coordinates of node F as (x_4, y_4). The length of line segment OF, denoted as l, is the distance between the obstacle and line segment KG along the longitudinal axis. The angle between line KG and the x-axis is α, whereas d denotes the distance from the obstacle to the path. The specific principle is as follows:

$$l = |y_3 - y_4|, \tag{3}$$

$$y_4 = \left(\frac{y_2 - y_1}{x_2 - x_1} (x_3 - x_1) + y_1 \right), \tag{4}$$

$$\alpha = \left| \arctan \frac{y_1 - y_2}{x_1 - x_2} \right|, \tag{5}$$

$$d = l \cos \alpha. \tag{6}$$

The distance d from the obstacle to the path is calculated following equation (6). Let D denote the safety threshold. Then, distance d is compared with the safety distance D to determine whether the path can be used as an alternative path. If $d \leq D$, the path is abandoned. Then, according to the improved A∗ algorithm, continue to search for new nodes until the optimal path is found. Otherwise (i.e., when $d > D$), the path is selected.
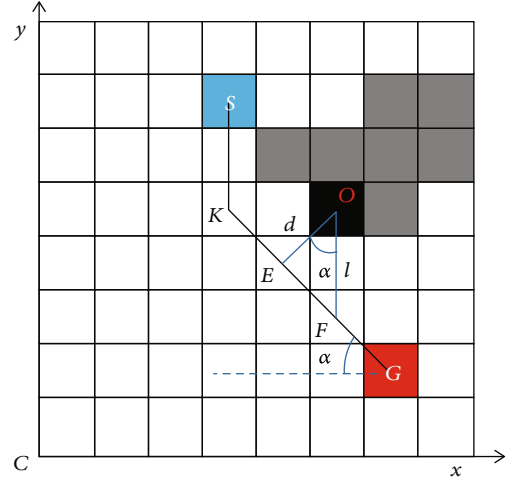

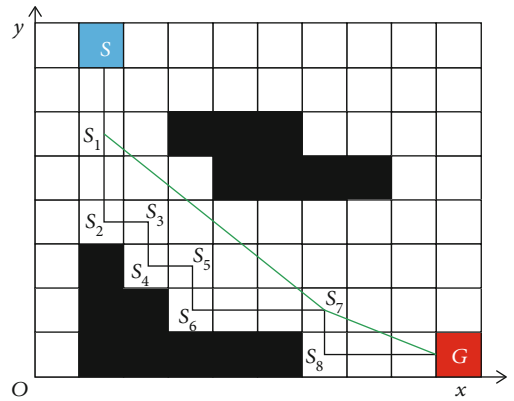
FIGURE 3: Safety distance design.

FIGURE 4: Floyd algorithm smoothing path comparison.

3.3. Path Smoothing Optimization. As emphasized previously, because the algorithm search principle and subnode search direction of standard A∗ algorithm are 8-node
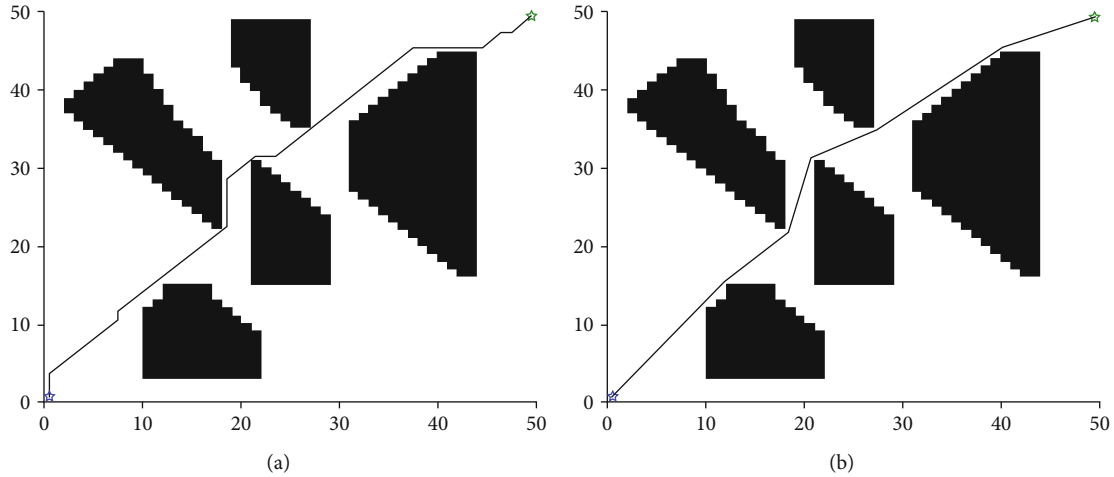
FIGURE 5: Comparison of the standard A∗ and the improved A∗ path planning results. (a) Standard A∗ algorithm. (b) Improved A∗ algorithm.

expansion direction, the path planned by the algorithm is not conducive to UAV flight. Further, path planning takes a long time, and the search route is close to the obstacle's edge. The improved A∗ algorithm combines the Floyd algorithm with the A∗ algorithm, effectively reducing the number of turning points in the UAV path [22], improving the UAV movement efficiency, and meeting the UAV application requirements. Considering the kinematic model to judge whether the derived path is more suitable than the original regarding distance, time, and angle, the algorithm optimization path comparison is shown in Figure 4.

As shown in Figure 4, the standard A∗ algorithm yields the path $(S, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, G)$. This path has many inflection points, resulting in poor smoothness.

The Floyd smoothing algorithm can be used to eliminate redundant path nodes, effectively reducing the number of turns and optimizing the path length. The Floyd algorithm combines the UAV motion characteristics to improve the path smoothness. By judging whether there is an obstacle between the two nodes, as well as considering the security threshold D and the distance d between the two nodes connected to the obstacle, one can determine whether the path is feasible. Finally, the path (S, S_1, S_7, G) is obtained.

Building on the described principles, the standard A∗ algorithm and the improved A∗ algorithm are simulated and verified, and the obtained paths are shown in Figure 5.

The path planned by the standard A∗ algorithm (Figure 5(a)) has many redundant nodes and road sections, and its turning angles are large. As a result, the path length is 75.1543 km, which is suboptimal (i.e., not the shortest) and does not conform to the UAV operation rules. In contrast, the path planned by the improved A∗ algorithm (Figure 5(b)) is relatively smooth, and its length equals 72.0735 km. Thus, the improved algorithm achieves a 4.1% improvement in the path length. Moreover, a certain safety distance can be observed between the global path and the obstacles, which is more conducive to the UAV movement.

4. Improved DWA

In the local path planning, the standard DWA algorithm only specifies the position of the target point, failing to consider unknown obstacles. Moreover, when there are many obstacles, the path planning algorithm easily falls into the local optimum, resulting in an increase in the global path length. The improved A∗ algorithm regards the global path's key nodes as subtarget points of the local path. Then, the environmental information is used to adjust the weight coefficient in the speed evaluation subfunction. Consequently, UAVs can successfully avoid unknown obstacles and reach the target point safely and efficiently.

4.1. Motion Model. The basic idea of DWA [6, 23, 24] is to predict the UAV's velocity vector space and state vector space at a particular time, simulate the UAV's trajectory at the predicted time, and finally select the optimal trajectory based on the evaluation function. Assuming that the UAV is moving in a uniform manner at every time interval Δt, the UAV kinematics model is as follows:

$$\begin{cases} x(t_n) = x(t_0) + \int_{t_0}^{t_n} V_r(t) \cos [\theta(t)] dt, \\ y(t_n) = y(t_0) + \int_{t_0}^{t_n} V_r(t) \sin [\theta(t)] dt, \\ \theta(t_n) = \theta(t_0) + \omega(t) \Delta t, \end{cases} \tag{7}$$

where $x(t_n), y(t_n)$, and $\theta(t_n)$ are the position coordinates and attitude angle coordinates of UAV in world coordinates at time t and $V_r(t)$ is the dynamic window (further introduced in the next section).

4.2. UAV Speed Sampling. DWA models the UAV's obstacle avoidance problem as an optimization problem with velocity constraints, including nonholonomic constraints, environmental obstacle constraints, and UAV dynamic constraints.

The DWA's velocity vector space diagram is shown in Figure 6. The abscissa and ordinate represent the UAV angular velocity (w) and the UAV linear velocity (v). In accordance, v_{max} and v_{min} denote the maximum and minimum linear velocity, whereas w_{max} and w_{min} are the maximum and minimum angular velocities. The whole area is V_s, the white area (V_a) represents the safe area, and V_d is the UAV speed range considering motor torque limitation in the control cycle. Finally, V_r is the UAV dynamic window determined by the intersection of the listed three sets.

According to the UAV speed limit, V_s is the set of the UAV's linear velocity and angular velocity fitting within the maximum dynamic window range:

$$V_s = \left\{ (v, w) \big|_{v_{min} \le v \le v_{max}, w_{min} \le w \le w_{max}} \right\}. \tag{8}$$

The UAV's motion trajectory can be subdivided into several linear or circular motions. In order to ensure the safety of UAV, under the condition of maximum deceleration, the current speed of UAV shall be able to hover in a certain accuracy range in the air before encountering obstacles, and the speed shall approach 0. The linear velocity and angular velocity set V_a near obstacles is defined as

$$V_a = \left\{ (v, w) \big|_{v \le \sqrt{2 dist(v,w)\dot{v}_b}, w \le \sqrt{2 dist(v,w)\dot{w}_b}} \right\}, \tag{9}$$

where $\dot{v}_b$ is the UAV's maximum linear deceleration, $\dot{w}_b$ is the UAV's maximum angular deceleration, and $dist(v, w)$ is the shortest distance between the path and the obstacle.

The UAV motor torque's limitations influence the maximum and minimum attainable velocities v and w in a certain time t. Thus, the dynamic window needs to be further reduced. Given the current linear velocity v_c and angular velocity w_c, the dynamic window V_d at the next period meets the following requirements:

$$V_d = \left\{ (v, w) \big|_{v_c - \dot{v}_b \Delta t \le v \le v_c + \dot{v}_a \Delta t, w_c - \dot{w}_b \Delta t \le w \le \dot{w}_a \Delta t} \right\}, \tag{10}$$

where $\dot{v}_a$ is the UAV's maximum linear acceleration and $\dot{w}_a$ is the UAV's maximum angular acceleration.

The final speed range is the set at the intersection of the three discussed sets, and the dynamic window V_r is defined as follows:

$$V_r = V_s \cap V_a \cap V_d. \tag{11}$$

The continuous velocity vector space (V_r) is discretized based on the number of linear velocity and angular velocity samples to obtain the discrete sampling points (v, w). For each sampling point, the UAV motion trajectory at the next moment is predicted using the UAV kinematics model (see Figure 7).

4.3. Adaptive DWA Algorithm. DWA controls the speed using the weight coefficient (β) in the speed evaluation subfunction. If β is too large, the UAV is very fast, but the safety is reduced. In contrast, if β is too small, the UAV's speed is insufficient, but the safety is very high. Considering the rela-
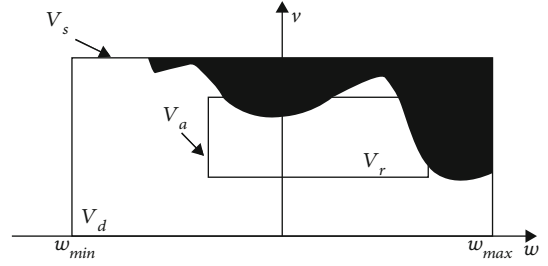

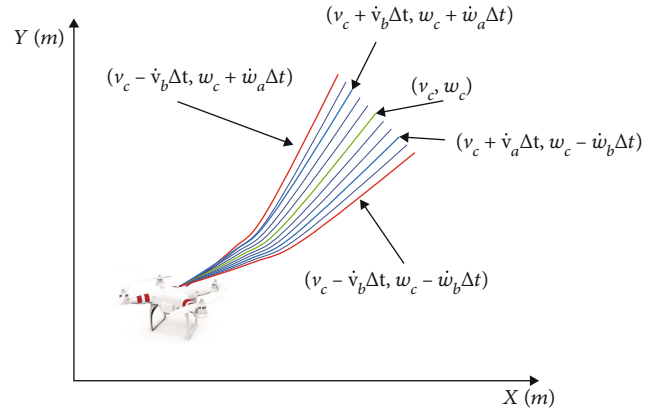FIGURE 6: Schematic diagram of the velocity vector space.

FIGURE 7: UAV trajectory prediction.

tionship between UAV safety and speed, the improved DWA algorithm adaptively calculates the speed evaluation function's weight coefficient while relying on environmental information. The UAV's sensors are utilized to collect environmental information. The obstacle density and the distance between the UAV and the obstacles in the environment are used to calculate the speed evaluation function's weight coefficient. The weight coefficient is adjusted to achieve higher speed in safe areas, whereas the speed is reduced in dangerous areas to enhance UAV flight safety.

4.3.1. Safety Threshold for Obstacle Detection. The distance L between different obstacles is calculated using the distance and angle between obstacles and the UAV. The specific calculation formula is as follows:[4]

$$L = \sqrt{D_i^2 + D_j^2 - 2 D_i D_j \cos(\theta)}, \quad D_i \le D_j, \tag{12}$$

where D_i and D_j are the distances between i-th and j-th obstacles and the UAV and θ is the angle between the UAV and the obstacles.

The distance between different obstacles serves to judge whether the UAV can pass through the obstacles. When L equals more than twice the two obstacles' expansion distance, the UAV can pass between the obstacles.

4.3.2. Adaptive Function Calculation. The UAV safety depends on the shortest distance between the UAV and the obstacle (denoted D_{min}), which serves as the input for
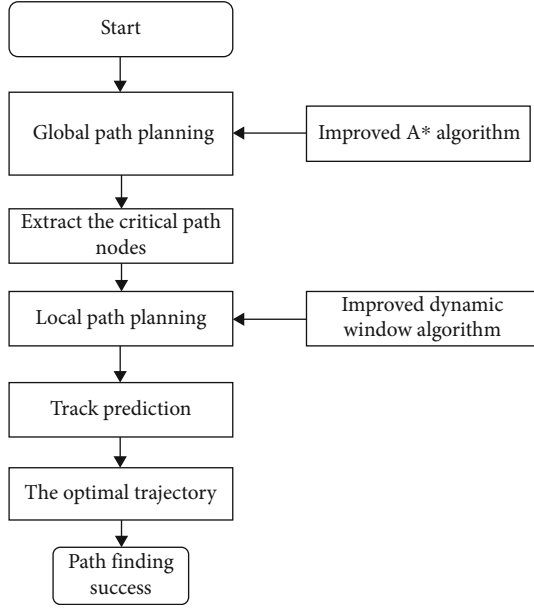
Figure 8: The fusion algorithm's flowchart.

Figure 9: The initial UAV airspace environment.

the adaptive dynamic function. The threshold D_l is defined as

$$D_l = n \cdot \left(\frac{v_{max}}{\dot{v}}\right), \tag{13}$$

where v_{max} is the UAV's highest speed, $\dot{v}$ is the UAV's linear acceleration, and n is the parameter. D_l is directly proportional to the maximum speed v_{max} and inversely proportional to the linear acceleration $\dot{v}$. Thus, the stronger the UAV's braking force, the smaller the threshold. The faster the speed, the larger the threshold, ensuring the UAV's safety.

The relationship between the speed evaluation function's weight coefficient β_r and input D_{min} is as follows:

$$\beta_r = \left\{ \beta_0 \cdot \eta e^{D_{min}/D_s}, \quad D_{min} \le D_s, \beta_0, D_{min} > D_s. \right. \tag{14}$$

If the UAV flight is within the safety threshold (i.e. the shortest distance D_{min} between the UAV and the obstacle is less than the UAV safety threshold, the UAV flight speed was reduced (according to the nature of exponential function). If the UAV flies outside the safety threshold (i.e., the shortest distance D_{min} between the UAV and obstacles is greater than the UAV safety threshold), the UAV flight speed is set according to the maximum speed weight, as shown in equation (14).

β_0 is the maximum speed weight, D_s is the safety threshold, and η is the adjustment parameter. When D_s is greater than D_{min}, the speed weight increases monotonically with the input D_{min}, i.e., $\beta_r = \beta_0$. When D_s is less than D_{min}, the speed weight is an exponential function of D_s and D_{min}. As the ratio of D_{min} and D_s decreases, β_r increases. Otherwise, if D_{min}/D_s increases, β_r decreases.
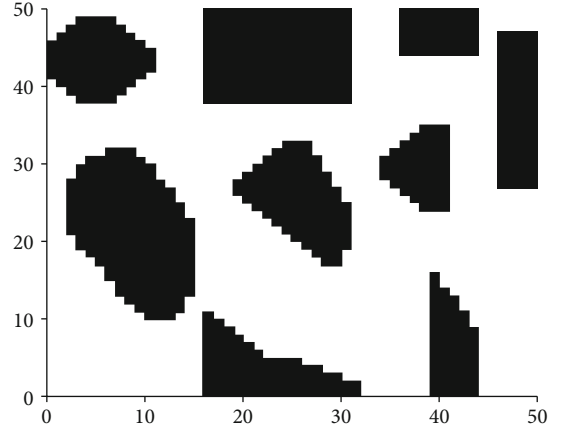
4.3.3. Improved Evaluation Function.
The standard DWA algorithm does not refer to the UAV's global path but only plans the target points. Moreover, in the case of numerous obstacles, the process easily falls into a local optimum, yielding a larger path distance. The dist(v, w) weight in the evaluation function moves the UAV trajectory away from obstacles, resulting in a locally optimal trajectory and increasing the global moving distance. If the corresponding weighting coefficient (γ) is directly reduced, the UAV faced with the unknown obstacles cannot avoid them in time, leading to the collision.

To solve the described problems, the azimuth evaluation function heading(v, w) is designed, which integrates the global path's nodes. Let vel(v, w) denote the evaluation function of the current simulation speed. Two different obstacle evaluation functions, dist_s(v, w) and dist_d(v, w), are designed. dist_s(v, w) is defined as the evaluation function for predicting the shortest distance between the end point of the trajectory and the globally known obstacle. Similarly, dist_d(v, w) is the evaluation function for predicting the shortest distance between the end point of the trajectory and the unknown obstacle. The trajectory evaluation function is as follows:

$$G(v, w) = \alpha \cdot \text{heading}(v, w) + \beta_r \cdot \text{vel}(v, w) + \gamma \cdot \text{dist\_s}(vw) \\ + \lambda \cdot \text{dist\_d}(v, w), \tag{15}$$

where α, β_r, γ, and λ are the coefficients of the four evaluations.

5. Fusion Algorithm

This work combines global and local path planning. The key nodes on the global path are regarded as the subtarget points of the local dynamic path to ensure the dynamic path planning's global optimality. The global path planning is carried out on the known map. The UAV's trajectory information in the next period is calculated based on the UAV's current linear speed and angular speed. Each UAV trajectory is predicted using the evaluation criteria, and the unreasonable trajectories or those where an obstacle is encountered are

TABLE 1: Path planning results when the obstacles are known.

Algorithm	Path length (km)	Time (s)	Speed increase rate
Improved A∗ algorithm	73.9423	0.2405	
Standard DWA algorithm	77.8257	3427.267	
Fusion of the improved A∗ and DWA algorithms	75.6582	2531.638	31.6%

TABLE 2: Path planning results when the obstacles are unknown.

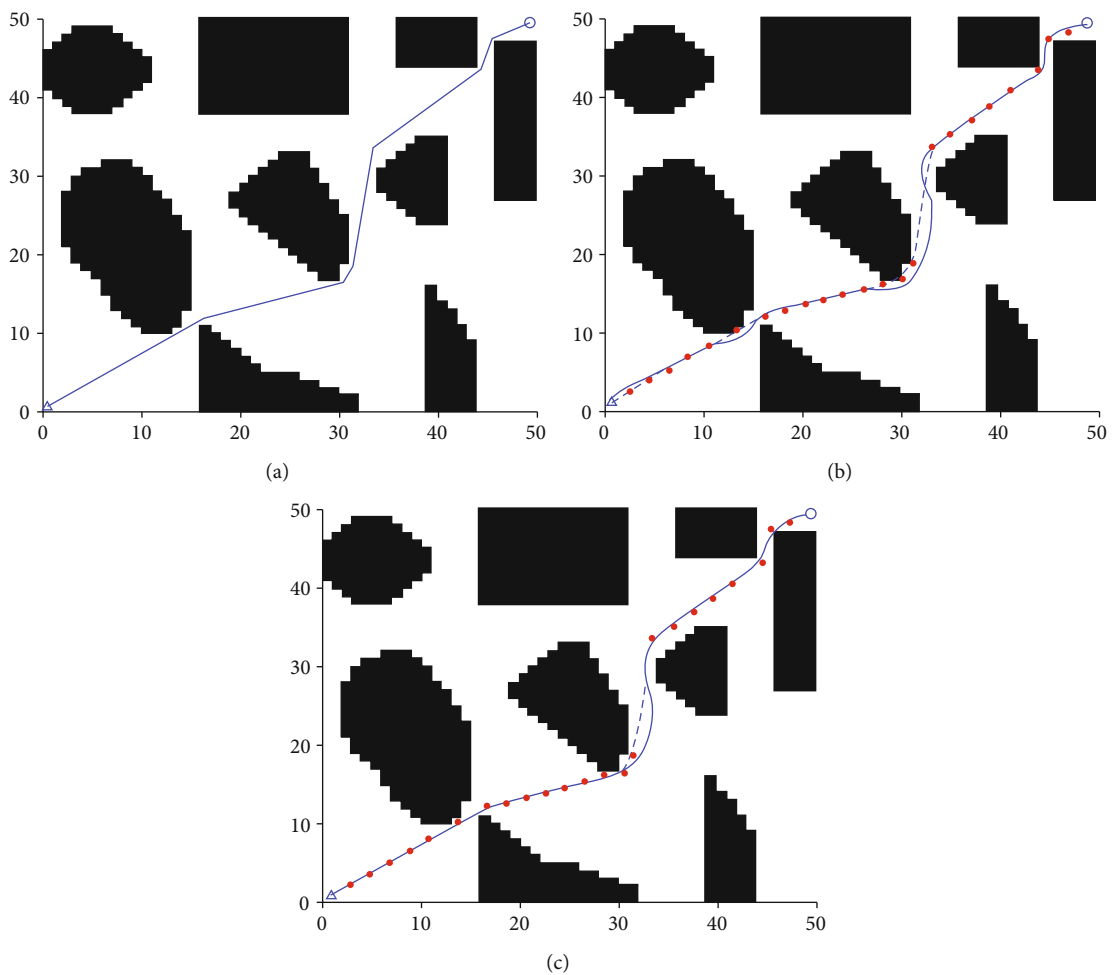| Algorithm	Path length (km)	Time (s)	Speed increase rate
Improved A∗ algorithm	73.9423	0.2405	
Standard DWA algorithm	78.5025	3023.206	
Fusion of the improved A∗ and DWA algorithms	76.5843	1942.824	34.2%

(a)

(b)

(c)

FIGURE 10: Comparison of the path planning algorithms on known maps. (a) Improved A∗ algorithm. (b) Standard DWA algorithm. (c) Fusion of improved A∗ and DWA algorithms.

discarded. The path evaluation subfunction evaluates a certain path, and the evaluation values are accumulated. The trajectory with the lowest cost is deemed as the optimal trajectory. If the current optimal trajectory is passable, the UAV moves with the speed corresponding to the optimal trajectory. The UAV avoids the encountered obstacles based on the obstacle avoidance rules. The flowchart of the improved algorithm is shown in Figure 8.

6. Simulation Experiment and Analysis

6.1. Experiment Environment. The main simulation environment consisted of an Intel (R) Core I i5-8265U CPU @1.60GHz, Windows10, and the simulations were conducted in MATLAB. A 50 km × 50 km path planning area and the corresponding grid map were established to verify the algorithm's effectiveness. The UAV's maximum turning
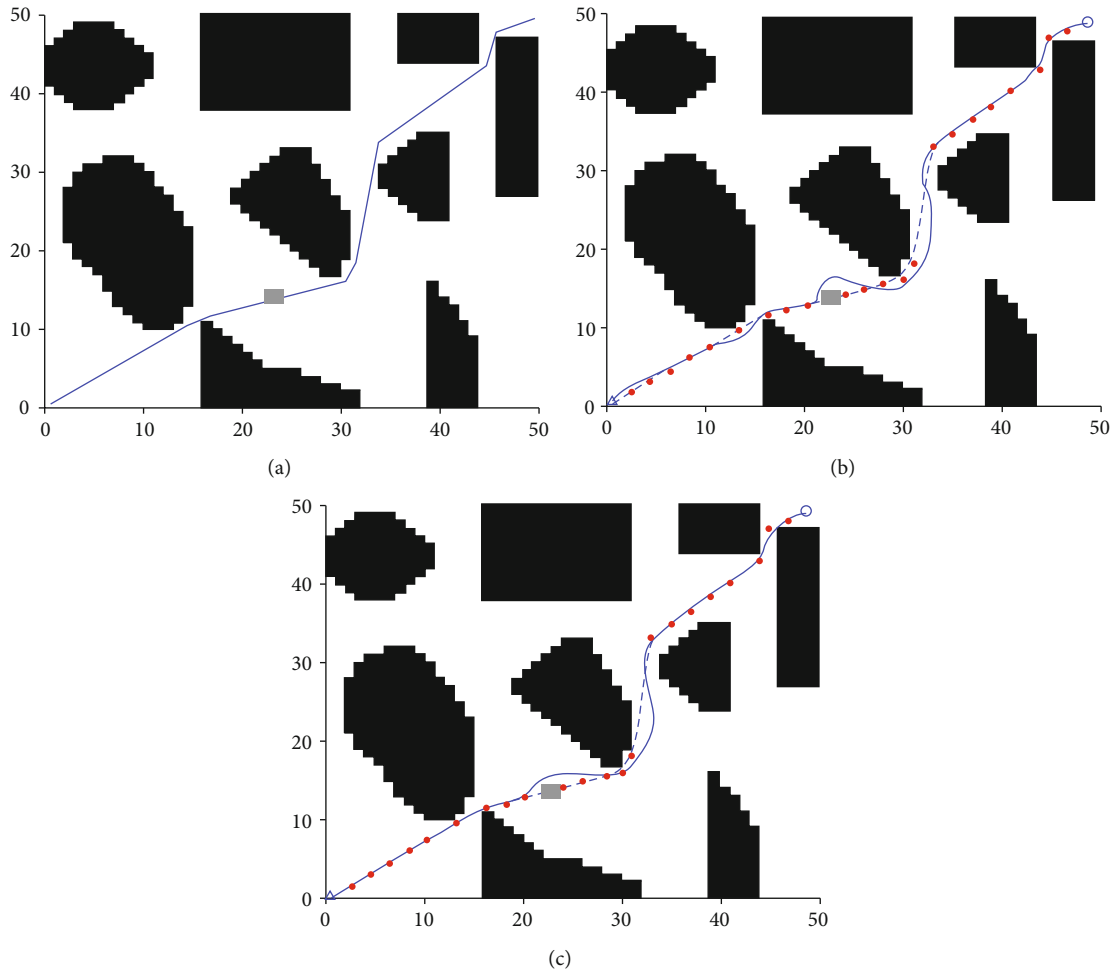
FIGURE 11: Comparison of the path planning algorithms on an unknown map. (a) Improved A∗ algorithm. (b) Standard DWA algorithm. (c) Fusion of improved A∗ and DWA algorithm.

angle was assumed to equal $\pi/4$. In the simulation, the grid granularity was set so that $N = 1$ km, the safety threshold equaled 0.8 km, and the UAV's starting and ending coordinates were (0.5, 0.5) and (49.5, 49.5). The UAV's flight environment is obtained by grid processing of arc obstacles and convex processing of concave obstacles. The UAV's initial airspace is shown in Figure 9.

6.2. Simulation Studies on Path Planning with Known and Unknown Obstacles

6.2.1. Path Planning in the Initial Airspace Environment. The simulation experiment compared the improved A∗ algorithm, the standard DWA algorithm, and the fusion of A∗ and DWA algorithms to obtain the planning time and the resulting path's length. The results are shown in Table 1.

The simulation results (Table 2 and Figure 10) demonstrate that the improved A∗ algorithm plans a globally optimal path. Compared with the improved A∗ algorithm, the path planned using the standard DWA algorithm is smoother but not globally optimal. Further, the planned path is not based on the planned global path, meaning there are several redundant paths. In Figure 10(c), the DWA algo-

rithm combines the global path planning and considers the environmental information during the UAV's flight. Consequently, the weight coefficient of the speed evaluation subfunction is adaptively adjusted, effectively reducing the UAV path planning time. Simultaneously, the obstacles are distinguished to reduce the impact of the known obstacles on the evaluation function, and the planned path is short and close to the global planning path.

According to Table 1, the length of the path planned using the fusion of improved A∗ and DWA algorithm is reduced by 2.8% compared to that obtained via the standard DWA algorithm. Further, the planning time is reduced by 26.3%, and the average UAV's flight speed is increased by 31.6%.

6.2.2. Path Planning When Obstacles Are Unknown. To verify the proposed algorithm's path planning performance in an unknown environment, obstacles are temporarily set on the path planned by the UAV. The gray box represents the unknown obstacles (see Figure 11). As can be seen from Figure 11, the improved A∗ algorithm cannot yield a path avoiding the unknown obstacle. The standard DWA algorithm successfully avoids the obstacles, but the UAV only
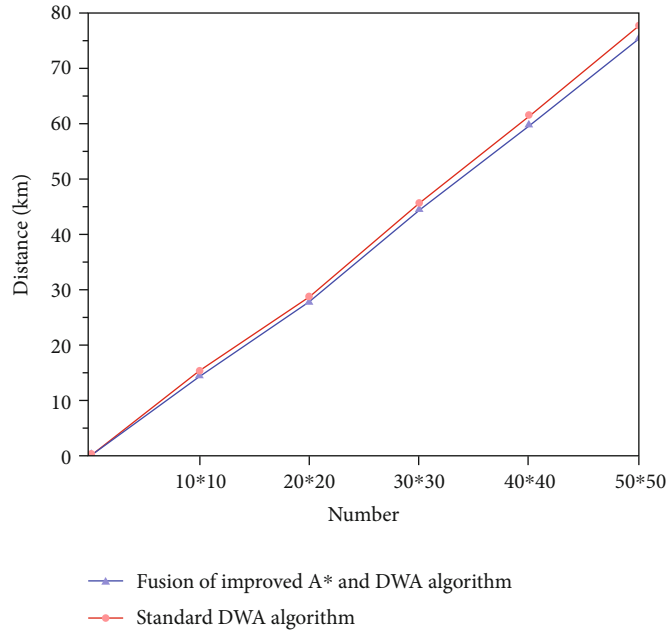
FIGURE 12: The relationship between the map size and the length of the paths yielded by the two algorithms.
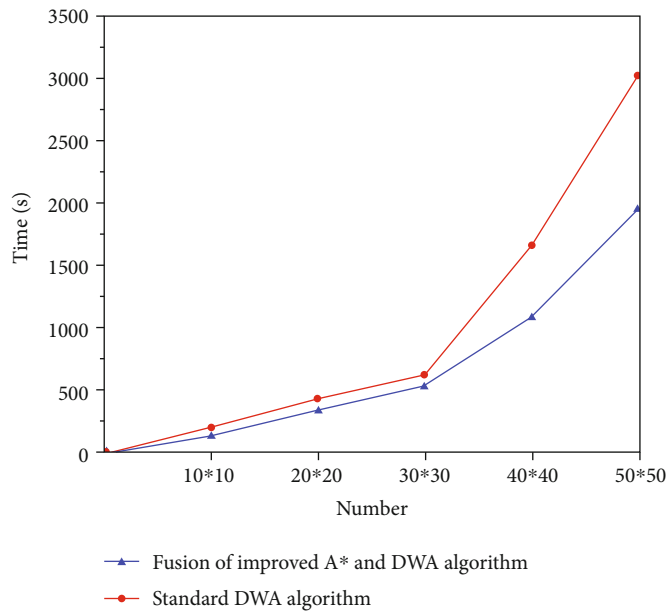

FIGURE 13: The relationship between the map size and the planning time planning of the two algorithms.

starts to bypass the obstacles when near them, enhancing the probability of a collision. In the case depicted in Figure 11(b), the UAV cannot fly along the initially planned global path, and there are many redundant paths, increasing the UAV's flight distance. By combining the improved A∗ algorithm and DWA algorithm, the local path planning can be realized, and the unknown obstacles can be avoided effectively. The local path is similar to the global path, proving that a smooth and safe path can be obtained.

Similar to the experiment reported in Section 6.2.1, the simulation experiment compared the improved A∗ algo-

rithm, the standard DWA algorithm, and the fusion of A∗ and DWA algorithms. The obtained path planning time and the path length are shown in Table 2.

As seen in Table 2, the improved A∗ and DWA algorithm reduces the planned path's length by 2.4% compared with the standard DWA algorithm. Further, the planning time decreases by 35.7%, whereas the average UAV's flight speed increases by 34.2%.

Next, environmental maps of different sizes were created to obtain environments of different complexity. The map ranges were set to $10 * 10$, $20 * 20$, $30 * 30$, $40 * 40$, and

50 ∗ 50. The shortest path length and path planning time under different environmental maps were extracted (Figures 12 and 13).

As shown in Figure 12, the increase in map size results in an increase in the UAV's flight path. Compared with the standard DWA algorithm, the average path planned by the fusion of the improved A∗ and DWA algorithms is reduced by 4.22%. The advantages of the improved A∗ and DWA algorithm become more pronounced with the increase in map size. Figure 13 shows that the path planning time of the fused improved A∗ and DWA algorithms is significantly shorter than that of the standard DWA. In addition, the increase in map size highlights the advantages of the proposed improved A∗ and DWA algorithm, signaling that the algorithm is more conducive to the UAV's path planning in a wide range.

7. Conclusion

This work alleviates the shortcomings in terms of smoothness and security of the path planned using the standard A∗ algorithm and tackles the problem of the standard DWA algorithm regarding its proneness to falling into a local optimum. The paper proposes a path planning algorithm that combines the improved A∗ algorithm and the DWA algorithm, leveraging the two algorithms' advantages and reducing their limitations. The main conclusions are as follows:

(1) Through grid modeling of an irregular map, this work establishes grid processing of arc obstacles and convex processing of concave obstacles

(2) Environment information and motion constraints are utilized to improve the A∗ algorithm. When compared to the standard A∗ algorithm, the path planned by the improved A∗ algorithm is smoother, and the UAV's motion safety is guaranteed

(3) An adaptive DWA is designed to dynamically adjust the path evaluation function using the safety threshold, UAV information, and environmental information regarding the obstacles

(4) The key nodes in the global path planned by the improved A∗ algorithm serve as the subtargets for the local path planning. Such a design ensures the optimal global path and improves the UAV's real-time path planning efficiency and safety

(5) The experimental results show that the path planning based on the fusion of improved A∗ and DWA algorithms effectively reduces the path length, shortens the UAV's path planning time, and improves both the path smoothness and the safety of the local obstacle avoidance

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

References

[1] M. M. Attoyibi, F. E. Fikrisa, and A. N. Handayani, "The implementation of a star algorithm (A∗) in the game education about numbers introduction," in *Proceedings of the 2nd International Conference on Vocational Education and Training (ICOVET 2018)*, Beijing, China, 2019.

[2] R. Kala, A. Shukla, and R. Tiwari, "Fusion of probabilistic A∗ algorithm and fuzzy inference system for robotic path planning," *Artificial Intelligence Review*, vol. 33, no. 4, pp. 307–327, 2010.

[3] L. W. Santoso, A. Setiawan, and A. K. Prajogo, *Performance Analysis of Dijkstra, A∗ and Ant Algorithm for Finding Optimal Path Case Study: Surabaya City Map, [Ph.D.Tthesis]*, Petra Christian University, 2010.

[4] M. Kothari, I. Postlethwaite, and D. W. Gu, "A suboptimal path planning algorithm using rapidly-exploring random trees," *International Journal of Aerospace Innovations*, vol. 2, no. 1, pp. 93–104, 2010.

[5] A. Özdemir and V. Sezer, "Follow the gap with dynamic window approach," *International Journal of Semantic Computing*, vol. 12, no. 1, pp. 43–57, 2018.

[6] J. Moon, B.-Y. Lee, and M.-J. Tahk, "A hybrid dynamic window approach for collision avoidance of VTOL UAVs," *International Journal of Aeronautical and Space Sciences*, vol. 19, no. 4, pp. 889–903, 2018.

[7] L. Tang, S. Dian, G. Gu, K. Zhou, S. Wang, and X. Feng, "A novel potential field method for obstacle avoidance and path planning of mobile robot," in *2010 3rd International Conference on Computer Science and Information Technology*, Chengdu, China, 2010.

[8] S. Y. Zhang, Y. K. Shen, and W. S. Cui, "Path planning of mobile robot based on improved artificial potential field method," *Applied Mechanics and Materials*, vol. 644-650, pp. 154–157, 2014.

[9] T. S. Rao, "An ant colony TSP to evaluate the performance of supply chain network," *Materials Today: Proceedings*, vol. 5, no. 5, pp. 13177–13180, 2018.

[10] D. Lyu, Z. Chen, Z. Cai, and S. Piao, "Robot path planning by leveraging the graph-encoded Floyd algorithm," *Future Generation Computer Systems*, vol. 122, pp. 204–208, 2021.

[11] T. Turker, O. K. Sahingoz, and G. Yilmaz, "2D path planning for UAVs in radar threatening environment using simulated annealing algorithm," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, Denver, CO, USA, 2015.

[12] P. Yao and H. Wang, "Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle," *Soft Computing*, vol. 21, no. 18, pp. 5475–5488, 2017.

[13] Z. Xu, R. Wei, K. Zhou, Q. Zhang, and R. He, "Laguerre graph self-optimize path planning algorithm for UAVs in irregular obstacle environment," in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, China, 2016.

[14] M. Boulares and A. Barnawi, "A novel UAV path planning algorithm to search for floating objects on the ocean surface based on object's trajectory prediction by regression," *Robotics and Autonomous Systems*, vol. 135, article 103673, 2021.

[15] A. Belkadi, H. Abaunza, L. Ciarletta, P. Castillo, and D. Theilliol, "Distributed path planning for controlling a fleet of UAVs : application to a team of quadrotors **research supported by the Conseil Regional de Lorraine, the Ministére de L'Education Nationale, de l'Enseignement Supérieur et de La Recherche, and the National Network of Robotics Platforms (ROBOTEX), from France," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15983–15989, 2017.

[16] L. Huang, H. Qu, P. Ji, X. Liu, and Z. Fan, "A novel coordinated path planning method using k -degree smoothing for multi-UAVs," *Applied Soft Computing*, vol. 48, pp. 182–192, 2016.

[17] D. Yulong, X. Bin, C. Jie et al., "Path planning of messenger UAV in air-ground coordination," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8045–8051, 2017.

[18] L. Huo, J. Zhu, Z. Li, and M. Ma, "A hybrid differential symbiotic organisms search algorithm for UAV path planning," *Sensors*, vol. 21, no. 9, 2021.

[19] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Applied Soft Computing*, vol. 107, article 107376, 2021.

[20] V. Jeauneau, L. Jouanneau, and A. Kotenkoff, "Path planner methods for UAVs in real environment," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 292–297, 2018.

[21] Z. Yuan, H. Chen, T. Li, Z. Yu, B. Sang, and C. Luo, "Unsupervised attribute reduction for mixed data based on fuzzy rough sets," *Information Sciences*, vol. 572, pp. 67–87, 2021.

[22] J. Wang, Y. Sun, Z. Liu, P. Yang, and T. Lin, "Route planning based on Floyd algorithm for intelligence transportation system," in *2007 IEEE International Conference on Integration Technology*, Shenzhen, China, 2007.

[23] X. Ji, S. Feng, Q. Han, H. Yin, and S. Yu, "Improvement and fusion of A* algorithm and dynamic window approach considering complex environmental information," *Arabian Journal for Science and Engineering*, vol. 46, no. 8, pp. 7445–7459, 2021.

[24] D. Feng, L. Deng, T. Sun, H. Liu, H. Zhang, and Y. Zhao, *Local Path Planning Based on an Improved Dynamic Window Approach in ROS*, Springer, 2021.