

An improved fuzzy-controlled local path planning algorithm based on dynamic window approach

Aizun Liu¹  | Chong Liu²  | Lei Li² | Ruchao Wang² | Zhiguo Lu²

¹Aerospace Era Feihong Technology Company Ltd., Beijing, China

²School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China

Correspondence

Aizun Liu, Aerospace Era Feihong Technology Company Ltd., 100094 Beijing, China.

Email: liaizun123@163.com

Abstract

With the increasingly complex operating environment of mobile robots, the intelligent requirements of robots are getting higher and higher. Navigation technology is the core of mobile robot intelligent technology research, and path planning is an important function of mobile robot navigation. Dynamic window approach (DWA) is one of the most popular local path planning algorithms nowadays. However, there are also some problems. DWA algorithm is easy to fall into local optimal solution without the guidance of global path. The traditional solution is to use the key nodes of the global path as the temporary target points. However, the guiding ability of the temporary target points will be weakened in some cases, which still leads DWA to fall into local optimal solutions such as being trapped by a "C"-shaped obstacle or go around outside of a dense obstacle area. In a complex operating environment, if the local path deviates too far from the global path, serious consequences may be caused. Therefore, we proposed a trajectory similarity evaluation function based on dynamic time warping method to provide better guidance. The other problem is poor adaptability to complex environments due to fixed evaluation function weights. And, we designed a fuzzy controller to improve the adaptability of the DWA algorithm in complex environments. Experiment results show that the trajectory similarity evaluation function reduces algorithm execution time by 0.7% and mileage by 2.1%, the fuzzy controller reduces algorithm execution time by 10.8% and improves the average distance between the mobile robot and obstacles at the global path's danger points by 50%, and in simulated complex terrain environment, the finishing rate of experiments improves by 25%.

KEY WORDS

DWA algorithm, dynamic obstacle avoidance, fuzzy control, local optimal, local path planning, mobile robot, trajectory similarity

1 | INTRODUCTION

With the development of computer science, the research of mobile robots has been carried out at a higher level. Due to the intervention of bionics, mobile robots have developed more mobile forms besides the simple wheeled movement (Campion & Chung, 2008), such as two-legged walking robot (Du et al., 2015), crawling robot (Raibert et al., 2010), and snake-like robot (Klaassen & Paap, 2003), mobile

robots have developed more adaptability to a complex environment, which makes the work of mobile robots from simple cargo transportation to intervention in extreme environments, such as explosive ordnance disposal and antiterrorism, outdoor terrain exploration and outer space exploration, and so on (Rubio et al., 2019). Navigation technology is the core of mobile robot intelligent technology research (Xu et al., 2020), and path planning is the most important function of mobile robot navigation (Patle et al., 2019).

The process of mobile robot path planning can be divided into two parts: one is the global path planning and the other is the local path planning (Liu, Jiang, et al., 2022).

In Orozco-Rosas et al. (2022), the researchers utilize the artificial potential field (APF) method to improve the classical Q-learning performance in convergence to the optimal solution and propose a QAPF learning algorithm for path planning, which reached an improvement of 18.83% in path length for the online mode, an improvement of 169.75% in path smoothness for the offline mode, and an improvement of 74.84% in training time over the classical approach. In Orozco-Rosas, Picos et al. (2019), the researchers proposed a path planning algorithm based on membrane pseudo-bacterial potential field (MemPBPF). Compared with other algorithms, the MemPBPF algorithm achieves high-performance results for mobile robot path planning in complex and real environments. In Orozco-Rosas, Montiel et al. (2019), the researchers proposed a membrane evolutionary APF approach. Compared with APF-based path planning methods, it shows a better performance regarding path length.

Our research focuses on local path planning algorithm.

With the complication of mobile robots' operating environment, the multiobjective path planning algorithm has become the major direction of global path planning research. In Liu, Liu et al. (2022), we proposed a global path planning algorithm based on multiobjective genetic algorithm. We proposed a new objective function to evaluate the falling risk of the global path. This algorithm is the global path planning algorithm mainly used in the experiment of this research.

Local path planner can plan local paths online and in real time, and controls the robot to avoid static or dynamic obstacles in the environment (Hu et al., 2023), which has high flexibility and real-time performance.

The dynamic window approach (DWA) has been one of the most popular algorithms in local path planning due to the advantages of movement fluency (Mai et al., 2021) and obstacle avoidance ability.

However, the traditional DWA algorithm has the following disadvantages:

- (1) In the absence of global planning, the traditional DWA algorithm solves the problem according to the current robot speed and position information, and the result is likely to be locally optimal (Cai et al., 2022), which leads to the robot easily falling into the "C"-shaped obstacles combination, resulting in the failure of path planning, and the need to extricate itself from difficulties, resulting in the increase of time and mobile mileage and it may even lead to the falling of the robot in a complex terrain environment.
- (2) In the traditional DWA algorithm, the weights of each evaluation function are fixed, that resulting in a lack of adaptability to complex environments. If the speed evaluation function is given a high-weight, the path planned by the algorithm may be too close to a certain obstacle, and the safety is difficult to be guaranteed (Maroti et al., 2013). When the robot approaches the dense obstacle area in the map, the algorithm usually does not cross the area under the action of the safety evaluation function (Trautman et al., 2013), but drives the robot around the dense obstacle area, which will also increase the moving time and total mileage of the robot (Castillo et al., 2006).

And many improved DWAs have been proposed by researchers. In Maroti et al. (2013), by eliminating the local minimum in most cases, the researchers proposed a DWA algorithm with a target point distance function. This method improves the evaluation performance of DWA, but when the weight of the evaluation function is inappropriate, the robot will be trapped. In Bai et al. (2021), the researchers proposed an objective cost function based on the surrounding environment information of the robot. This method can avoid the DWA from falling into a local optimal to some extent. In Yan et al. (2022), the researchers proposed a new constraint on the distance from the current trajectory point to the target in the evaluation function, which improved the obstacle avoidance of the DWA algorithm. This algorithm lacks the guidance of the global path and may still fall into the local optimal. Fusion algorithm is also a major research direction of path planning algorithms study. In Yin et al. (2023), the researcher developed a fusion algorithm for the automatic guided vehicle; this algorithm combines the kinematical constraint A* algorithm with the DWA algorithm. However, this algorithm has few improvements for DWA, and the performance improvement of the DWA algorithm is not obvious. In Zhang et al. (2024), a hybrid algorithm based on improved A* and fuzzy control DWA is proposed. The researchers designed a DWA fuzzy controller that sets the distance of the vehicle from the target point G-dist (G) and the distance of the vehicle from the nearest obstacle O-dist (O) as inputs to adjust the weights of the DWA evaluation function, experiment results show that in three experimental environments, the average path length of the improved DWA is reduced compared to the DWA, but the improved DWA has a certain sacrifice in algorithm execution time. In Li et al. (2024), the researchers propose an A* and DWA fusion algorithm. The attitude adjustment function is introduced to optimize the DWA algorithm and eliminate the attitude deviation between the robot's initial pose and the current path pose direction, and a new evaluation function is proposed to take dynamic obstacles into consideration in local path planning. In Song et al. (2024), to solve the problem that the trajectory planned by DWA may not align closely with the global path, leading to unnecessary redundant changes in direction during the robot's movement, the researchers introduce an evaluation function representing the distance from the endpoint of the predicted trajectory to the target point and an evaluation function to represent the distance from the endpoint of the predicted trajectory to the next node on the global path.

Considering the disadvantages of the above algorithms and the actual needs of the current mobile robot path planning, an improved DWA algorithm is proposed in this paper. The main contribution of this paper is the development of a new local path planning algorithm based on DWA and fuzzy control method. The proposed algorithm can ensure that the local path is more consistent with the global optimal path, and also has excellent adaptability to the actual environment. The proposed algorithm has been tested extensively validated compared with an improved local path planning algorithm based on the classical DWA. Both planning algorithms are tested in benchmark environments and compared concerning measures, such as execution time, mileage, and distance to obstacles. The contributions of our research can be summarized as follows:

- (1) A new similarity evaluation function is proposed for the DWA algorithm, which can make local path closer to the global path and improve the phenomenon that DWA is prone to fall into local optimal.
- (2) A fuzzy controller is designed, which can dynamically adjust the weights of evaluation functions in DWA, to improve the adaptability of the algorithm to different environments.
- (3) The experiments and analysis demonstrate that the proposed fuzzy-controlled guided DWA (FGDWA) algorithm successfully solves that the traditional DWA algorithm that DWA is prone to local optimal and poor adaptability to complex environments.

The rest of this paper is organized as follows. In Section 2, the traditional DWA algorithm is introduced, and the proposed new evaluation function is explained, then the fuzzy controller is explained. In Section 3, the effectiveness of the proposed improved DWA algorithm is verified in the simulation environment. Section 4 is the conclusion. Section 5 is the prospect of further research.

2 | INSTRUCTION OF ALGORITHM

In this section, we will first briefly introduce the traditional DWA algorithm. Then systematically explain the improved method proposed in our research.

2.1 | Traditional DWA algorithm

DWA is an action sampling algorithm. According to the current speed and angular velocity information of the robot, the algorithm samples in the speed space to obtain multiple sets of speed, simulates the trajectory of the robot in the next time period according to a certain time interval, then evaluates the trajectory according to the evaluation function, selects the speed information corresponding to the optimal trajectory, and sends it to the robot to drive the robot to move. The quality of path planning usually depends on the quality of the initially constructed evaluation function (Lin et al., 2021). The flowchart of the DWA algorithm is shown in Figure 1.

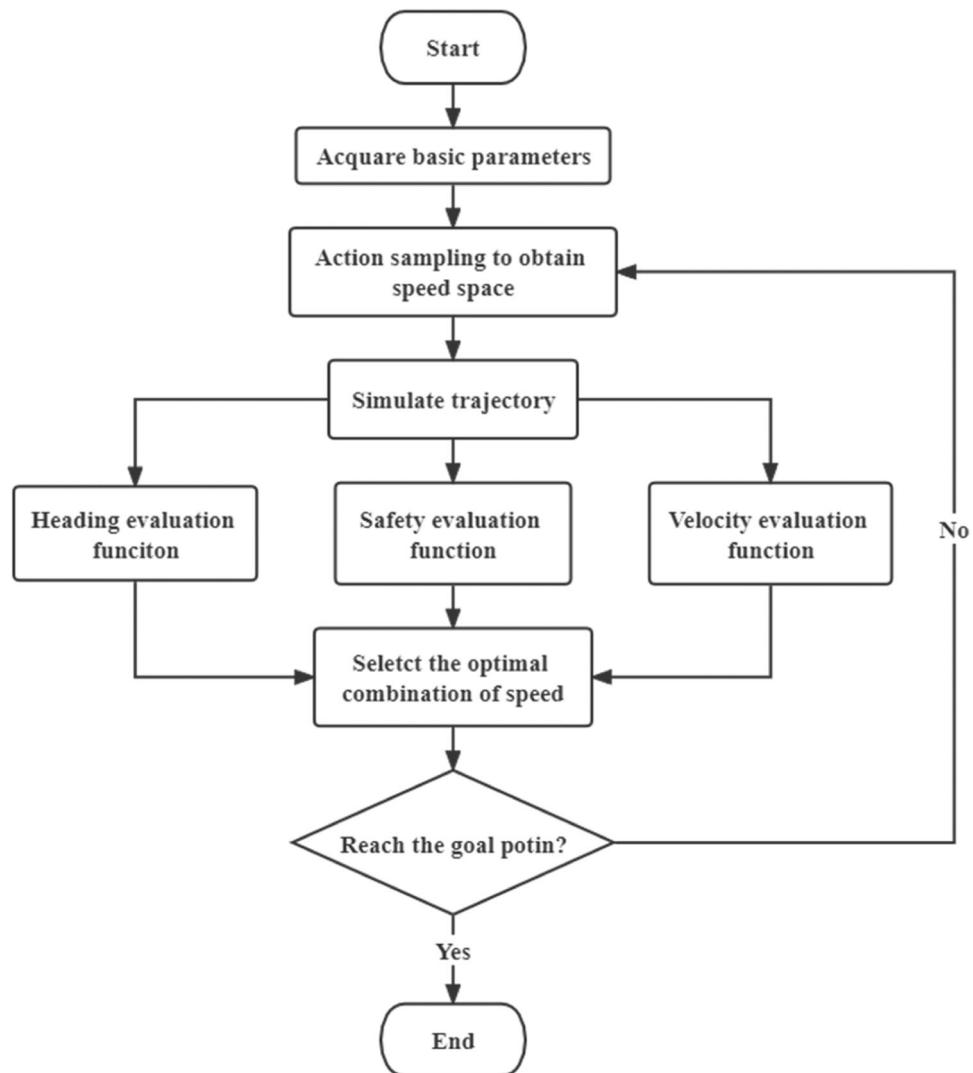


FIGURE 1 Flowchart of dynamic window approach algorithm.

2.1.1 | Mathematical model of robot movement

DWA algorithm needs to simulate the trajectory of the robot, so it needs to determine the mathematical model of robot movement. Next, we illustrate the algorithm principle based on a two-wheel differential mobility robot.

We assume that the robot can only move forward, not backward. When simulating the robot trajectory, the time interval is small, and the robot moving distance is short, so the robot trajectory between two moments can be approximately regarded as a straight line, as shown in Figure 2.

2.1.2 | Velocity space of robot

The DWA algorithm converts the control of the robot position into the control of the robot speed, and converts the obstacle avoidance problem into an optimization problem in the speed space with constraints, according to the current position and speed of the robot. There are two main types of constraints: (1) robot self-constraints and (2) robot safety constraints (Maroti et al., 2013).

The maximum speed and minimum speed limit of the robot motor, and the speed combination $V_m(v, \omega)$ satisfies, as shown in Equation (1).

$$V_m = \{v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\}. \quad (1)$$

Because the motor torque is limited and there is a maximum acceleration limit, there is a speed combination $V_d(v, \omega)$, as shown in Equation (2).

$$\begin{aligned} V_d = \{(v, \omega) | & v \in [v_c - \dot{v}_d \times dt, v_c + \dot{v}_d \times dt] \wedge \omega \\ & \in [\omega_c - \dot{\omega}_d \times \Delta t, \omega_c + \dot{\omega}_d \times \Delta t]\}. \end{aligned} \quad (2)$$

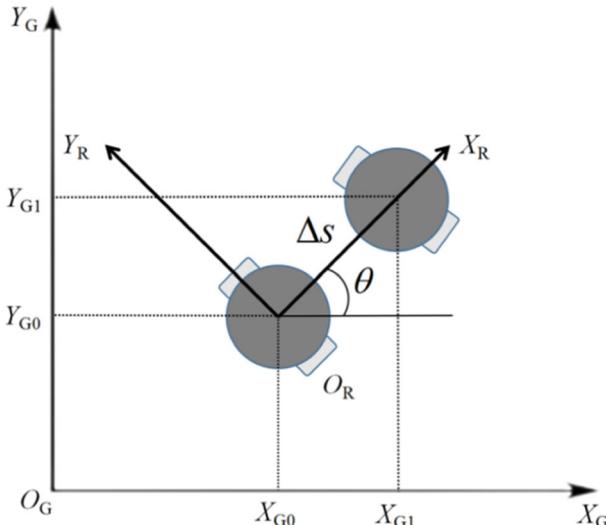


FIGURE 2 Changes of robot position at certain time intervals. [Color figure can be viewed at wileyonlinelibrary.com]

In this equation, v_c and ω_c are the linear velocity and angular velocity of the robot, \dot{v}_d and $\dot{\omega}_d$ are the acceleration, dt is the given time interval when simulating the trajectory forward.

To stop the robot in time before hitting obstacles, there is a speed combination $V_s(v, \omega)$, as shown in Equation (3).

$$\begin{aligned} V_s = \{& (v, \omega) | v < \sqrt{2 \times \text{dist}(v, \omega) \times \dot{v}_d} \\ & \wedge \omega < \sqrt{2 \times \text{dist}(v, \omega) \times \dot{\omega}_d}\}. \end{aligned} \quad (3)$$

In this equation, $\text{dist}(v, \omega)$ is the nearest distance from the robot to the obstacle under the speed combination (v, ω) .

2.1.3 | Evaluation function

After obtaining the speed space of the robot, the algorithm simulates the trajectory of the robot in a certain time according to a certain time interval, as shown in Figure 3, and selects the simulated trajectory through the evaluation function to obtain the optimal speed combination of the evaluation function.

The evaluation function comprehensively considers the heading, distance from obstacles and moving speed of the robot, which makes robot move towards the target point as quickly as possible on the premise of avoiding obstacles, as shown in Equation (4).

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega)). \quad (4)$$

In this equation, $\text{heading}(v, \omega)$, $\text{dist}(v, \omega)$, and $\text{velocity}(v, \omega)$ are the evaluation subfunctions, α , β , and γ represent the weight values of the three subfunctions, σ represents the normalization of the evaluation functions. The influence of weight value on algorithm effect is as follows:

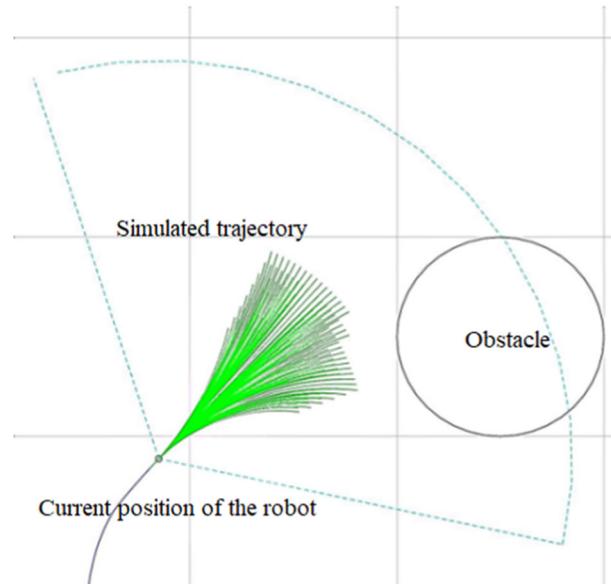


FIGURE 3 Simulation of robot trajectory. [Color figure can be viewed at wileyonlinelibrary.com]

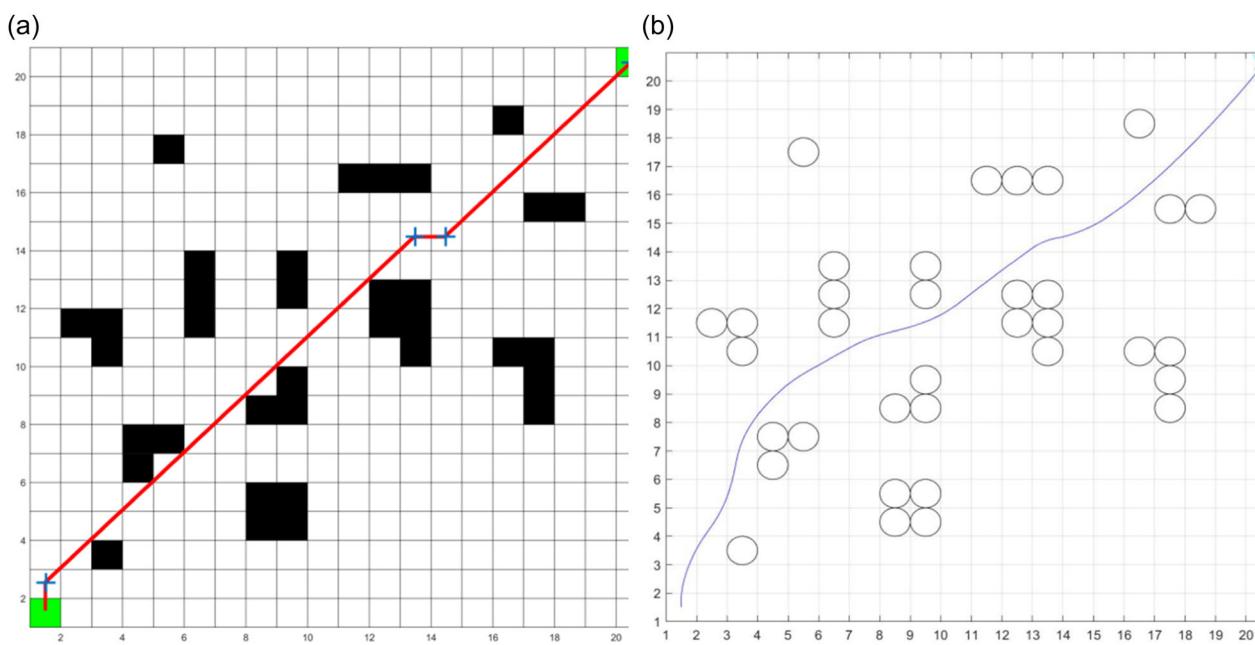


FIGURE 4 Result of GDWA with traditional evaluation functions. (a) Global path and (b) actual path of local path planning. GDWA, guided dynamic window approach. [Color figure can be viewed at wileyonlinelibrary.com]

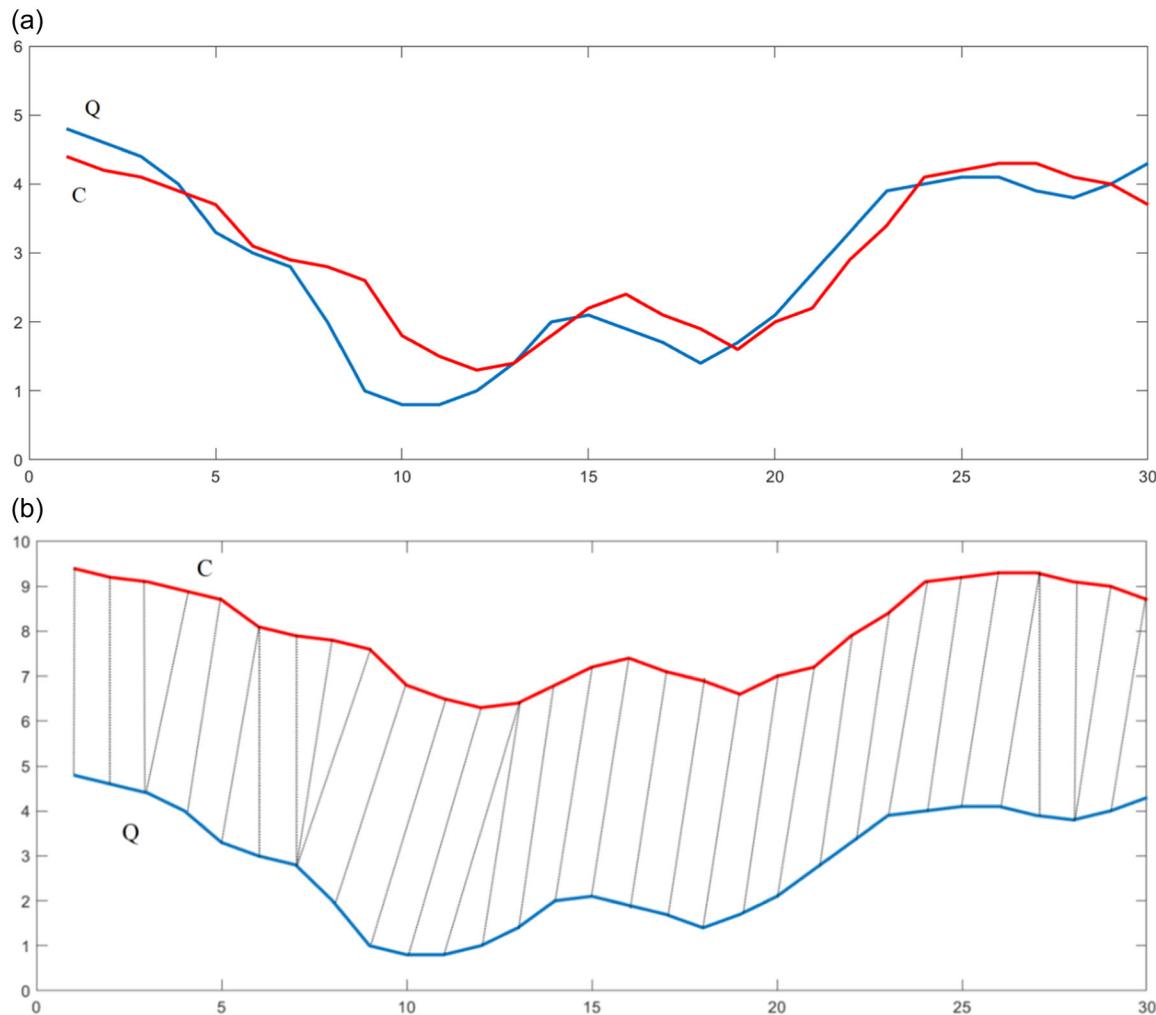


FIGURE 5 Sample sequence. (a) Original sequences and (b) lifted and aligned sequences. [Color figure can be viewed at wileyonlinelibrary.com]

When the weight of the heading evaluation function α is high, the robot will be more inclined to move towards the goal point, ignoring safety and moving speed and not considering whether it conforms to the global path. When it is low, it would cause the robot to give up moving towards the goal point.

When the weight of the safety evaluation function β is higher, the robot will be more active in avoiding obstacles, that may cause the robot to deviate from the heading of the goal point or have difficulty in approaching the goal point near the dense obstacles. When β is low, the robot may get too close to an obstacle and collide with it.

When the weight of the velocity evaluation function γ is high, the robot will ignore directionality and safety, and may not be able to reach the goal point or collide with obstacles. When it is low, the speed of robot would be low cause waste of time.

2.2 | Improvement of DWA

We made the following improvements for DWA: first, we propose a trajectory similarity evaluation function to make local path planning closer to the global path and improve the phenomenon that DWA is prone to fall into local optimal. Then a fuzzy controller is designed, which can dynamically adjust the weight of evaluation function of DWA algorithm, to improve the adaptability of the algorithm to different environments.

2.2.1 | DWA guided by the global path

To solve the problem that DWA algorithm is easy to fall into local optimal. The commonly used method is to use the key nodes of the

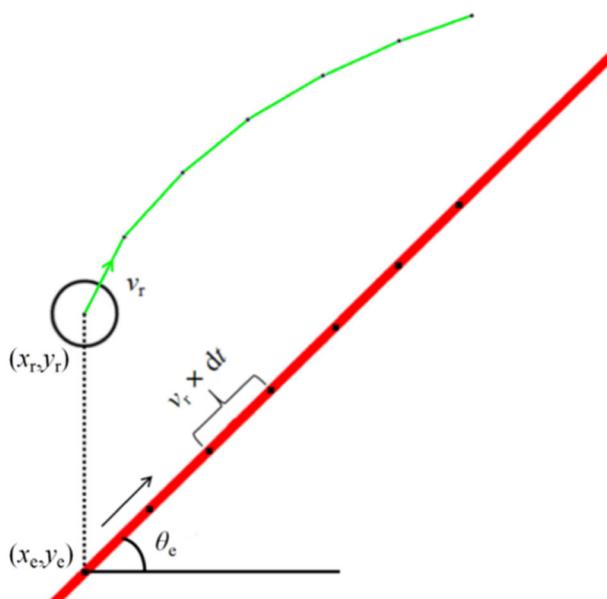


FIGURE 6 Sampling example. [Color figure can be viewed at wileyonlinelibrary.com]

global path obtained from global path planning as the temporary target points of the DWA algorithm (Yin et al., 2023). We called it guided DWA (GDWA).

However, when the interval between key nodes is too large, its guidance ability will be weakened. As shown in Figure 4a, there is a global path generated by the A* algorithm, and the key nodes are marked by “+”. Figure 4b shows the actual path obtained by DWA under the guidance of key nodes. It can be seen that there are two key nodes far apart in the global path, and the guidance ability of the latter key node is weak, which is because DWA has to avoid obstacles when planning local paths (Bai et al., 2021), and makes the actual path of DWA deviate from the global path.

To solve those problems above, we propose a new evaluation function: trajectory similarity evaluation function, which evaluates the similarity between simulated trajectory and global path, so as to strengthen the guidance ability of the global path.

Trajectory similarity evaluation method

Trajectory similarity measurement method. Spatiotemporal trajectory similarity, also known as trajectory similarity measurement methods, can be divided into two categories: trajectory point-based measurement and trajectory segment-based measurement (Wu et al., 2022). The moving trajectory of the robot is continuous, and it is very difficult to obtain and store the continuous trajectory. Therefore, in most cases, we use discrete points to represent the continuous trajectory, and the simulated trajectory of DWA is composed of multiple discrete trajectory points, so we adopt point-based measurement, including Euclid (Bao et al., 2016), Edit Distance on Real Penalty (Agrawal et al., 2005) and dynamic time warping (DTW) (Chen & Ng, 2009). DTW is a method of stretching or shrinking trajectories locally to compare trajectories with different sampling rates and lengths. It does not require

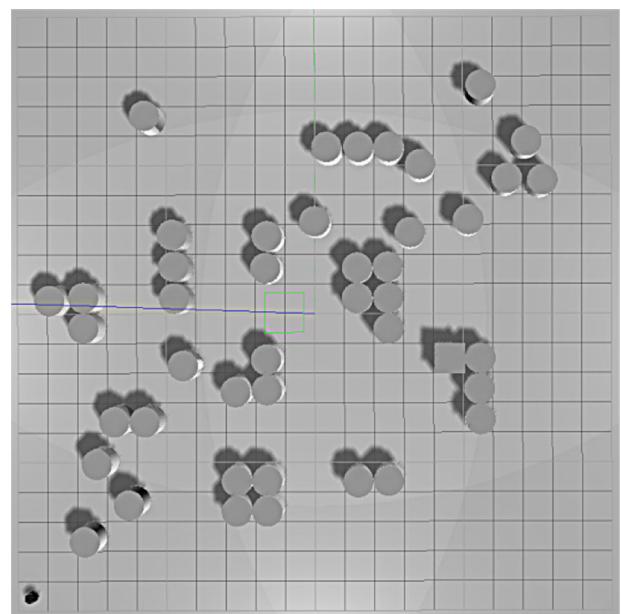


FIGURE 7 Global map with dynamic obstacles.

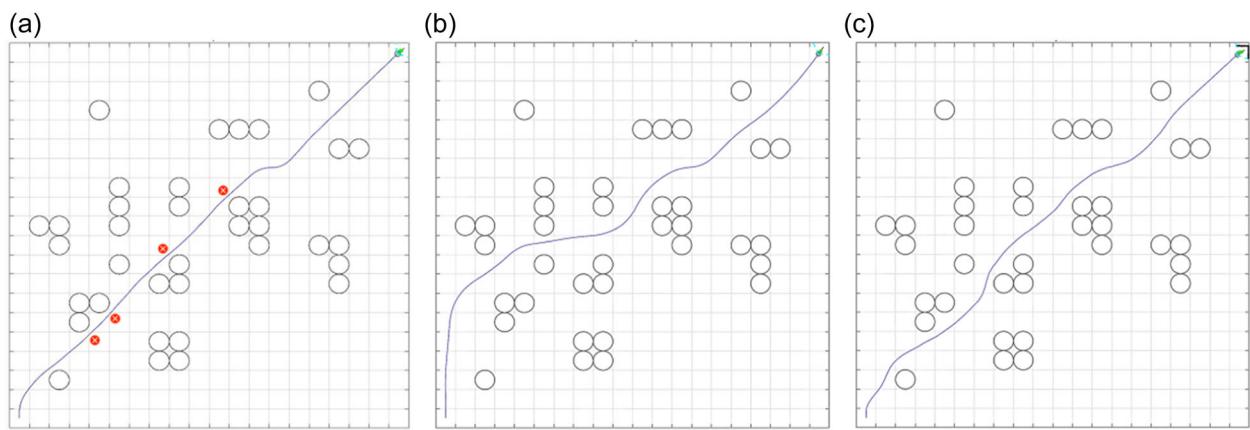


FIGURE 8 Path results of guided dynamic window approach algorithm under different fixed weights. (a) Similarity function with high-weight, (b) safety function with high-weight, and (c) velocity function with high-weight. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

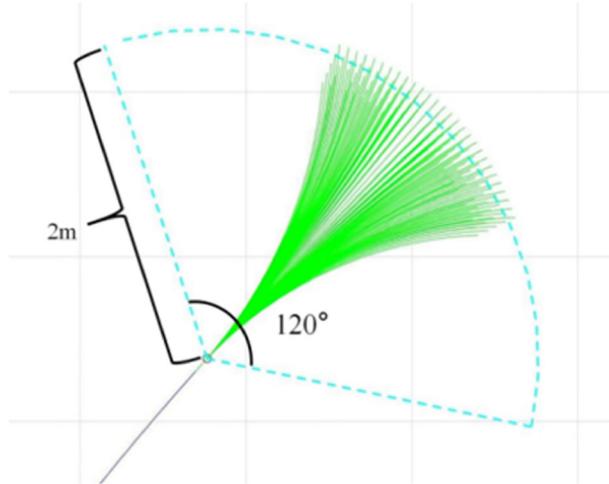


FIGURE 9 Range of detect window. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

one-to-one correspondence of trajectory points, but directly uses the distance between trajectory points and their matching points as a part of the similarity results, so it has good real-time calculation. Therefore, DTW is used as the evaluation method of trajectory similarity in our research.

Dynamic time warping. Suppose there are two one-dimensional time series Q and C of length n and m , respectively. As shown by Equation (5),

$$\begin{cases} Q = q_1, q_2, q_3, \dots, q_i, \dots, q_n, \\ C = c_1, c_2, c_3, \dots, c_i, \dots, c_m. \end{cases} \quad (5)$$

Taking the sequences Q and C shown in Figure 5a as an example, we can see that their shapes are very similar, but they are not aligned in time. Therefore, before calculating the similarity, we need to warp them under the time axis to achieve a better alignment effect. DTW is to warp the time axis to align the two sequences Q and C , to obtain the minimum distance $\text{minDist}(q_i, c_j)$ of all corresponding points

between the two sequences. Since the sample sequence is one-dimensional data, the distance $\text{Dist}(q_i, c_j)$ between two points is defined here as shown in Equation (6)

$$\text{Dist}(q_i, c_j) = |q_i - c_j|. \quad (6)$$

Each matrix element (i, j) represents the alignment of points q_i and c_j .

After obtaining the minimum distances of all corresponding points, all the minimum distances are summed to be the DTW distances of the current two sequences. Figure 5b is the warped sequences.

Trajectory similarity evaluation function

Global path sampling. DWA algorithm is based on the current position and velocity of the robot to simulate the local path trajectory information. So the simulated trajectory does not need secondary processing. However, the acquisition of the global path does not consider the actual speed and position of the robot, and it is a set of coordinates without the concept of time. So we need to resample the global path to obtain the global path with time information. The specific processing steps are as follows:

(a) Normally, the original information of the global path is a set of point coordinates, and its image is a broken line. After segmenting the broken line, we calculate the functional expression of each broken line segment, assuming that the equation of a certain global path is as shown in Equation (7).

$$ax + by + c = 0. \quad (7)$$

(b) The original information on the global path does not contain directions. We divide the global path into segments and calculate the direction of each segment, that is, the angle between the path direction and the positive direction of the x -axis, and set it as θ_e .

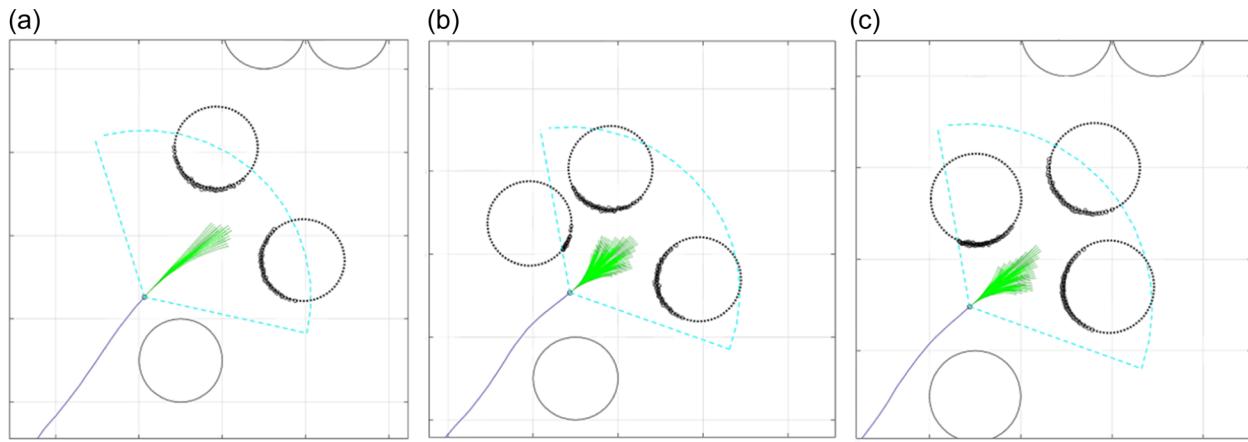


FIGURE 10 Example of density. (a) Density $I = 0.7753$, (b) density $I = 1.0802$, and (c) density $I = 1.1401$. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Input and output subsets of the fuzzy controller.

| Variate | Fuzzy subset |
|----------|--|
| Input | |
| I | Small (S), medium (M), large (L) |
| D | Small (S), medium (M), large (L) |
| Output | |
| β | Small (S), medium (M), large (L) |
| δ | Small (S), medium (M), large (L) |

TABLE 2 Input and output domain of the fuzzy controller.

| Variate | Domain |
|----------|----------|
| Input | |
| I | $[0, 2]$ |
| D | $[0, 5]$ |
| Output | |
| β | $[0, 1]$ |
| δ | $[0, 5]$ |

TABLE 3 Output β rule of the fuzzy controller.

| β | I | S | M | L |
|---------|-----|-----|-----|-----|
| $D = S$ | S | M | L | |
| $D = M$ | S | M | L | |
| $D = L$ | S | M | M | |

TABLE 4 Output δ rule of the fuzzy controller.

| δ | I | S | M | L |
|----------|-----|-----|-----|-----|
| $D = S$ | S | S | S | |
| $D = M$ | M | M | M | |
| $D = L$ | L | L | M | |

(c) After obtaining the function expression of the global road and the path direction, we sample the global path, and calculate the current ideal position (x_e, y_e) of the robot according to the current position information (x_r, y_r) of the robot, taking the x -axis as the standard, as shown in Equation (8).

$$\begin{cases} x_e = x_r, \\ y_e = \frac{-c}{\alpha x_r}. \end{cases} \quad (8)$$

According to the current speed information v_r of the robot, we simulate the trajectory information and the sample points will be carried out at a time interval of one dt , as shown in Equation (9).

$$\begin{cases} x_i = x_e + i \times v_r \times dt \times \cos(\theta_e), \\ y_i = y_e + i \times v_r \times dt \times \sin(\theta_e). \end{cases} \quad (9)$$

In this equation, i represents the number i sampling point.

As shown in Figure 6, taking a simulated trajectory as an example, the green line is the DWA simulated trajectory, the red line is the global path, and the black points on the global path are the sampling points.

In this way, we obtain two sequences: the simulated trajectory of DWA and global path sampling points. Then we can measure the similarity by DTW, and calculate the trajectory similarity evaluation function.

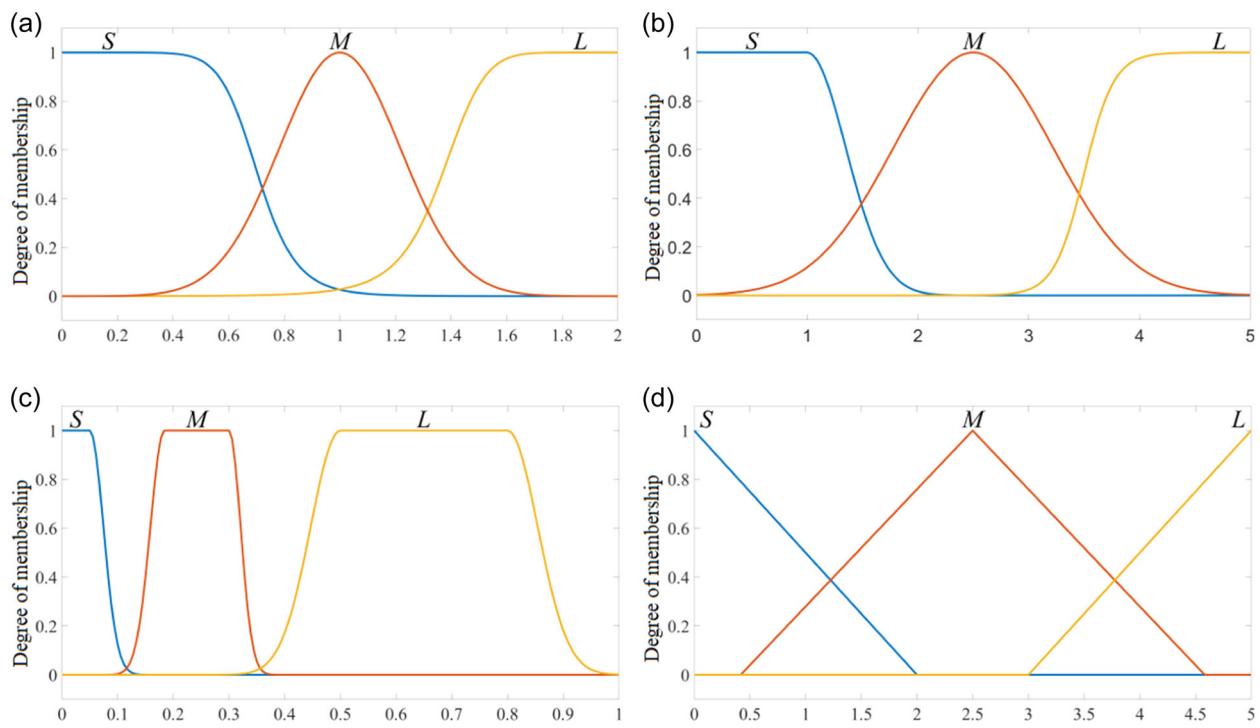


FIGURE 11 Images of each input and output membership function. (a) Image of input I membership function, (b) image of input D membership function, (c) image of output β membership function, and (d) image of output δ membership function. [Color figure can be viewed at wileyonlinelibrary.com]

Trajectory similarity evaluation function. In the original DTW, the comparison sequences are one-dimensional time series, and the distance calculation method is Equation (6). When DTW is applied to evaluate the similarity between the simulated trajectory and the global path, the sampling points set is a two-dimensional time series, and the original distance calculation method is no longer applicable. We redefine $\text{Dist}(p_r, p_s)$ as shown in Equation (10):

$$\text{Dist}(p_r, p_s) = \sqrt{(x_r - x_s)^2 + (y_r - y_s)^2}. \quad (10)$$

In this equation, p_r and p_s are the DWA trajectory points and global path sampling points, and the redefined distance $\text{Dist}(p_r, p_s)$ is the two-dimensional Euclidean distance between simulated trajectory points and global path sampling points.

After redefining the distance $\text{Dist}(p_r, p_s)$, we get the trajectory similarity evaluation function $\text{similarity}(R, E)$, as shown in Equation (11):

$$\text{similarity}(R, E) = \frac{1}{\text{DTW}(R, E) + 1}. \quad (11)$$

In this equation, R and E are the DWA's simulated trajectory sequence and global path sampling points sequence, $\text{DTW}(R, E)$ is the dynamic time warping process. To prevent $\text{DTW}(R, E)$ from being 0, the molecular part of the evaluation function is assigned an initial value of 1.

So we get a new DWA evaluation function as shown in Equation (12)

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega) + \delta \cdot \text{similarity}(v, \omega)). \quad (12)$$

The time complexity of traditional DWA algorithms is usually $O(n^2)$, where n is the size of the robot's state space, that is, the number of nodes in the simulated trajectory. The time complexity of DTW algorithms is usually $O(n^2)$, where n is the number of sequence nodes. Therefore, the time complexity of the algorithm proposed in our research is $O(n^3)$.

2.2.2 | Fuzzy adaptive parameter control

To improve the adaptability of DWA algorithm to complex environment under the fixed weight of evaluation functions, fuzzy controller is adopted to control the weight of evaluation functions in our research.

For example, the global map of the simulation environment shown in Figure 7 is roughly similar to the global map shown in Figure 4a, but there are obstacles that are not described in the grid map, so we think these obstacles are dynamic obstacles.

Under the condition of fixed weights, taking the global map path shown in Figure 4b as the benchmark, the results are shown in Figure 8. If the weight of the similarity evaluation function is relatively high, the robot is more inclined to follow the global path guidance, which may cause the robot to be too close to obstacles, as shown by the red mark, which may cause a collision and make safety difficult to guarantee; If the weight of safety evaluation function is

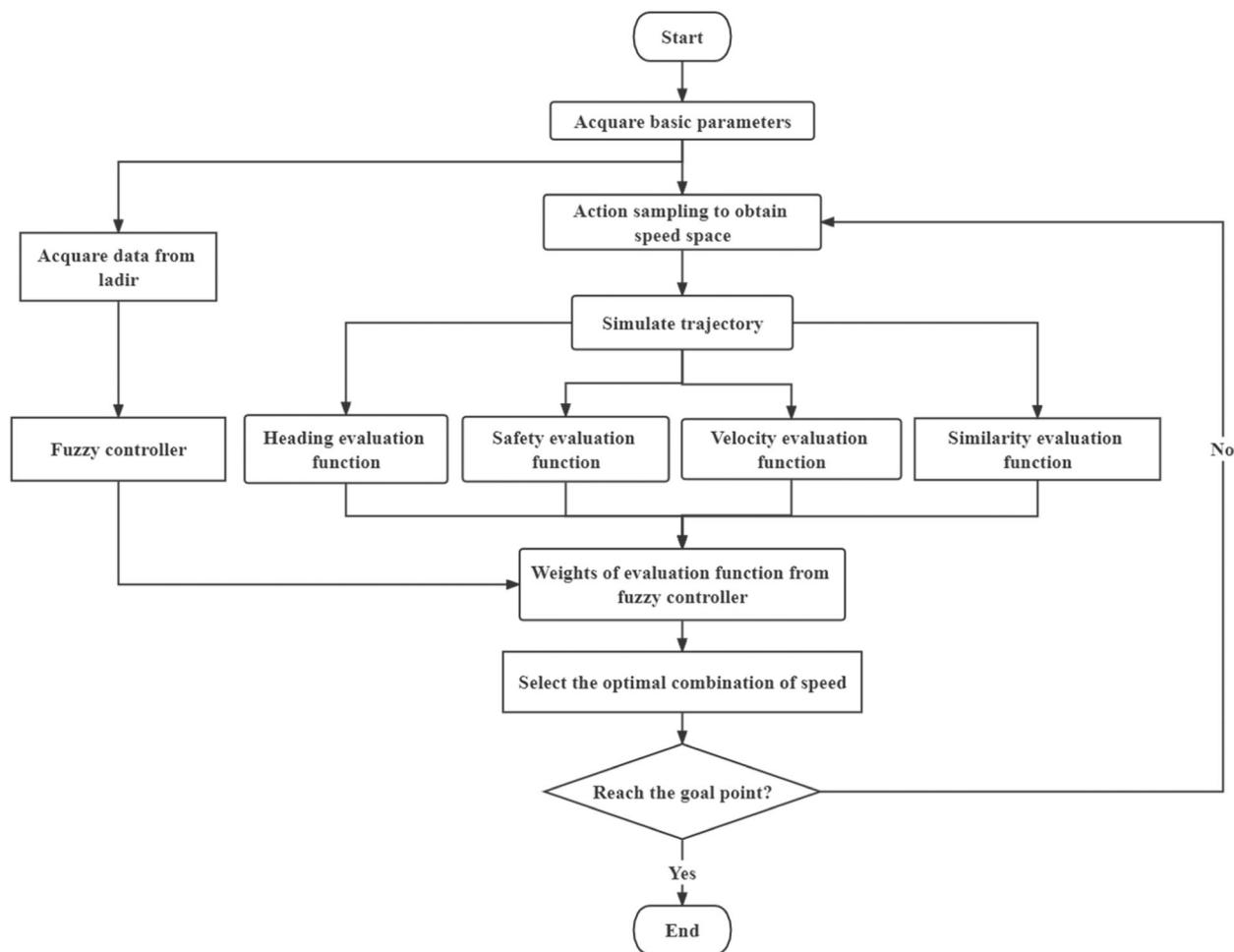


FIGURE 12 Flowchart of fuzzy-controlled guided dynamic window approach algorithm.

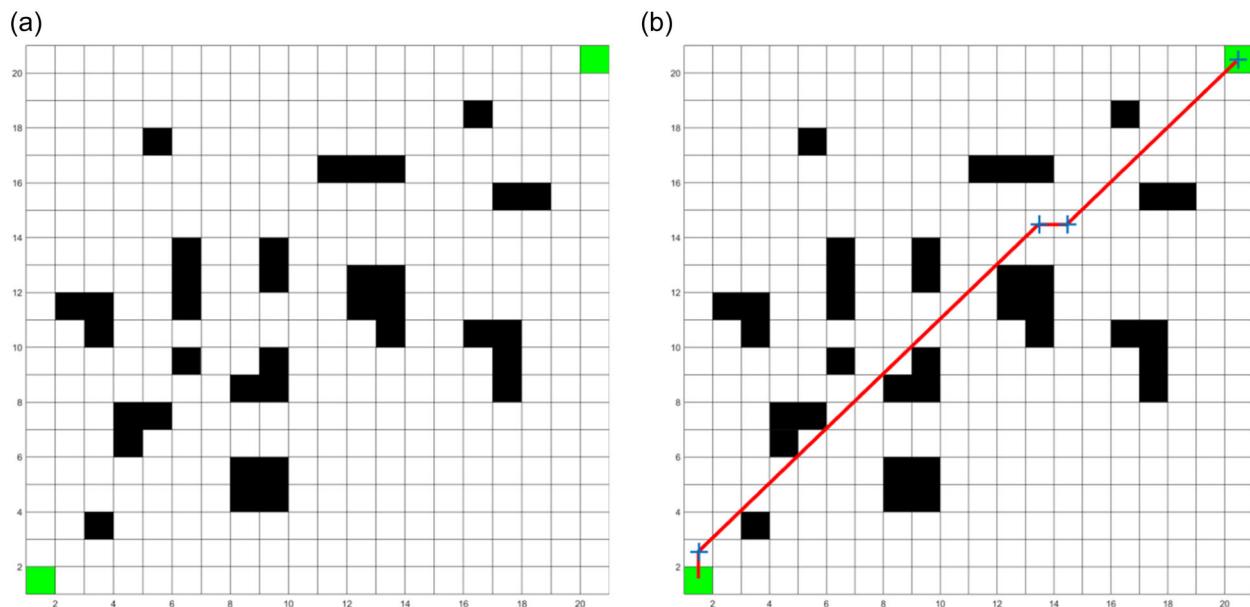


FIGURE 13 Experimental map and its global path. (a) Global map and (b) global path. [Color figure can be viewed at wileyonlinelibrary.com]

relatively high, the robot is more inclined to bypass obstacles, which causes the robot to deviate from the global path and fall into the local optimal problem again before encountering dynamic obstacles; If the weight of velocity evaluation function is relatively high, the robot tends to follow the simulated trajectory with the maximum linear velocity. When encountering dynamic obstacles and needing to avoid

TABLE 5 Parameters of evaluation functions.

| Weight of heading evaluation function α | Weight of safety evaluation function β | Weight of velocity evaluation function γ |
|--|--|---|
| 0.5 | 0.15 | 1 |

them, the speed may change suddenly, resulting in poor path smoothness. The reason for these problems is that once the weights of the DWA evaluation functions are determined, it cannot be adjusted in real time and is difficult to adapt to the actual working environment (Keogh & Ratanamahatana, 2005).

Detect window setting

In practical application, to identify dynamic obstacles, mobile robots need to obtain the real information of the surrounding environment through sensors. The sensor used in our research is lidar. Usually, the detection range of lidar is 270°. Consider the mobile robot motion model in our research, the sensor data can be sliced to improve the algorithm efficiency.

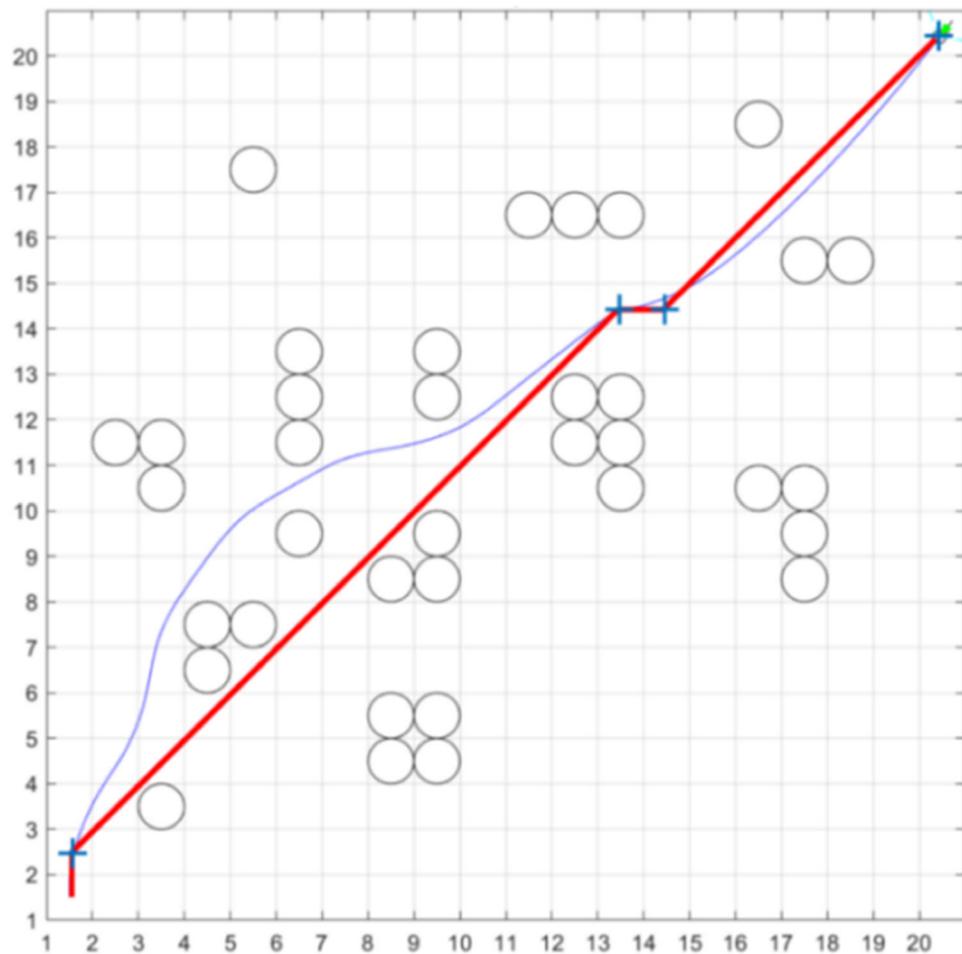


FIGURE 14 Guided dynamic window approach path without trajectory similarity evaluation function. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 6 Parameters of evaluation functions.

| Weight of heading evaluation function α | Weight of safety evaluation function β | Weight of velocity evaluation function γ | Weight of similarity evaluation function δ |
|--|--|---|---|
| 0.5 | 0.15 | 1 | 0.5 |

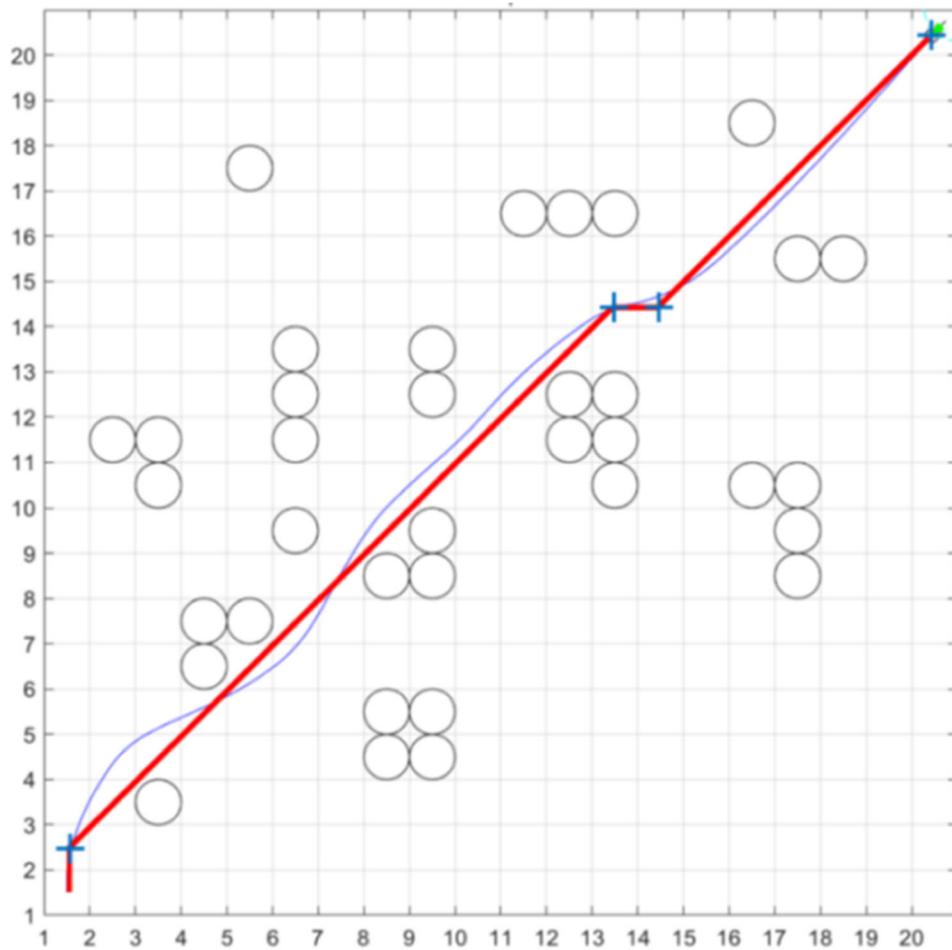


FIGURE 15 Guided dynamic window approach path with trajectory similarity evaluation function. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 7 Data comparison before and after algorithm improvement.

| Algorithm | Execution time (s) | Iteration step (r) | Mileage (m) |
|-----------------|--------------------|--------------------|-------------|
| GDWA | 157.6 | 361 | 28.2 |
| Similarity-GDWA | 156.5 | 352 | 27.6 |
| Effect (%) | 0.7 | 2.5 | 2.1 |

Abbreviation: GDWA, guided dynamic window approach.

TABLE 8 General parameters of the experiment.

| Weight of heading evaluation function α | Weight of velocity evaluation function γ |
|--|---|
| 0.5 | 1 |

As shown in Figure 9, the sector formed by blue dotted lines is the detect window, and the angle range of the detect window in our research is 120° and the radius is 2 m.

Parameter calculation

Through lidar, the robot can obtain the information of obstacles in the detect window. If the control algorithm is used to adjust the parameters of DWA adaptively, it is necessary to set parameters as inputs. In our research, two parameters are inputs of the fuzzy controller: (1) obstacle density I in the sensing window and (2) the deviation D of the current robot from the global path.

Obstacle density I. We define the obstacle density I in the detect window as shown in Equation (13):

$$I = \frac{S_d}{S_w} + c^{\max_gap}. \quad (13)$$

In this equation, S_d is the range of obstacles scanned by lidar, S_w is the range of the whole detect window, and the quotient of these two is the proportion of obstacles occupying the detect window; c is the base coefficient of exponential function, which we set to 0.5, and \max_gap is the maximum distance of obstacles in the detect window. It can be deduced that under ideal conditions, the minimum density I is 0, and the maximum density I is 2.

As shown in Figure 10, the obstacle distribution and obstacle density I in these three cases are 0.7753, 1.0802, and 1.1401. Experiments show that the calculation method represented by Equation (13) can effectively express the obstacle density in the detect window.

Path deviation D. We define the distance from the current robot position to the global path as the path deviation D . The minimum of D is 0. According to the robot model studied in this paper, we set the maximum of D to 5, which means the maximum deviation is only calculated to 5 m. If it exceeds 5 m, it is considered that the deviation distance of the robot is too far and beyond the adjustment range.

Adaptive parameter fuzzy controller

Fuzzy control is based on fuzzy mathematics and consists of fuzzy set theory, fuzzy language, and fuzzy logic. It belongs to nonlinear intelligent control, that can transform people's thinking into fuzzy logic (Hao et al., 2019), to control objects effectively if precise models cannot be established. From application experience, fuzzy control has advantages, such as achieving multiobjective control, expert control, and strong robustness (Tanaka & Sugeno, 1992).

The fuzzy controller designed in this paper has a dual-input and dual-output structure. The input parameters are obstacle density I

and path deviation D , and the output values are DWA evaluation function weights β and δ . The fuzzy subsets of each input and output are shown in Table 1.

The domains of input and output are shown in Table 2. Output β is the weight of safety evaluation function. In actual experiments, we find that DWA is sensitive to the change of β , so taking its domain as $[0, 1]$ can meet the adjustment requirements; Output δ is the weight of trajectory similarity evaluation function, and its domain is $[0, 5]$.

The fuzzy control rules of each input and output variable are shown in Tables 3 and 4.

The membership function of each input and output variable is shown in Figure 11.

After the completion of the fuzzy controller, we complete the FGDWA. The flowchart of the algorithm is shown in Figure 12.

3 | EXPERIMENT

In this section, we conducted experiments and analyzed the results to verify the effectiveness of the proposed trajectory similarity evaluation function and the fuzzy controller designed in our research. Then, we integrated the local path planning algorithm proposed in our research with the global path planning algorithm proposed by Liu,

TABLE 9 Running time of 15 experiments.

| Running time (s) | FGDWA | Safety evaluation function with high-weight | Similarity evaluation function with high-weight |
|------------------|-------|---|---|
| 1st | 86.64 | 113.81 | 94.58 |
| 2nd | 89.0 | 120.04 | 100.18 |
| 3rd | 87.39 | 124.59 | 101.87 |
| 4th | 88.74 | 127.14 | 107.52 |
| 5th | 95.5 | 133.58 | 115.95 |
| 6th | 87.76 | 109.19 | 86.65 |
| 7th | 88.12 | 113.51 | 89.96 |
| 8th | 88.82 | 115.08 | 90.28 |
| 9th | 87.49 | 119.74 | 91.75 |
| 10th | 88.57 | 120.04 | 99.34 |
| 11th | 88.89 | 122.07 | 99.74 |
| 12th | 87.11 | 122.59 | 105.13 |
| 13th | 94.28 | 123.05 | 105.88 |
| 14th | 87.72 | 125.18 | 106.24 |
| 15th | 97.7 | 130.73 | 110.54 |
| Average | 89.58 | 121.36 | 100.37 |
| Effect (%) | - | 26.2 | 10.8 |

Abbreviation: FGDWA, fuzzy-controlled guided dynamic window approach.

TABLE 10 Iteration step of the 15 experiments.

| Iteration step (r) | FGDWA | Safety evaluation function with high-weight | Similarity evaluation function with high-weight |
|--------------------|--------|---|---|
| 1st | 170 | 207 | 173 |
| 2nd | 170 | 223 | 175 |
| 3rd | 170 | 221 | 177 |
| 4th | 171 | 222 | 187 |
| 5th | 171 | 233 | 202 |
| 6th | 171 | 203 | 169 |
| 7th | 169 | 211 | 170 |
| 8th | 173 | 213 | 170 |
| 9th | 170 | 221 | 172 |
| 10th | 169 | 221 | 186 |
| 11th | 173 | 223 | 186 |
| 12th | 171 | 227 | 197 |
| 13th | 172 | 228 | 197 |
| 14th | 173 | 235 | 199 |
| 15th | 177 | 241 | 199 |
| Average | 171.33 | 221.93 | 183.93 |
| Effect (%) | - | 22.8 | 6.8 |

Abbreviation: FGDWA, fuzzy-controlled guided dynamic window approach.

Liu et al. (2022), and conducted experiments in simulated environment to verify the effectiveness of the algorithm.

The simulation experiment is conducted by MATLAB 2020b, the configuration of computer is Core i5 central processing unit (3.8 GHz), 8 GB random-access memory (RAM), and the Windows 10 system.

TABLE 11 Mileage of 15 experiments.

| Mileage (m) | FGDWA | Safety evaluation function with high-weight | Similarity evaluation function with high-weight |
|-------------|-------|---|---|
| 1st | 27.35 | 27.90 | 27.29 |
| 2nd | 27.35 | 31.74 | 27.27 |
| 3rd | 27.34 | 31.59 | 27.36 |
| 4th | 27.37 | 27.84 | 27.38 |
| 5th | 27.36 | 29.86 | 27.34 |
| 6th | 27.34 | 28.21 | 27.28 |
| 7th | 27.33 | 29.73 | 27.27 |
| 8th | 27.36 | 28.05 | 27.27 |
| 9th | 27.34 | 28.11 | 27.39 |
| 10th | 27.34 | 28.03 | 27.28 |
| 11th | 27.37 | 27.97 | 27.29 |
| 12th | 27.33 | 28.27 | 27.37 |
| 13th | 27.44 | 28.89 | 27.31 |
| 14th | 27.35 | 32.31 | 27.39 |
| 15th | 27.38 | 29.88 | 27.38 |
| Average | 27.36 | 29.22 | 27.32 |
| Effect (%) | - | 6.4 | -0.1 |

Abbreviation: FGDWA, fuzzy-controlled guided dynamic window approach.

3.1 | Experiment of the proposed evaluation function

First, we verify the validity of the proposed similarity evaluation function. We compare GDWA (without similarity evaluation function) with GDWA (with similarity evaluation function). Figure 13 shows the global map used in the experiment and the global path.

In the evaluation function of the traditional DWA algorithm, the three fixed weights are generally obtained empirically and experimentally (Zhou et al., 2023). Therefore, we set experience values for the weight of GDWA evaluation functions without similarity evaluation function, as shown in Table 5. In Section 2.2.2 we have discussed the effect of increasing the weight value of the safety evaluation function and the velocity evaluation function.

As shown in Figure 14, we can see that it deviates from the global optimal path.

Table 6 shows the weights of GDWA (with similarity evaluation function) evaluation functions. The result is shown in Figure 15. We can see that the local path is in good agreement with the global optimal path.

A quantitative comparison of data before and after algorithm improvement is shown in Table 7. We can see that the GDWA algorithm with similarity evaluation function is more consistent with the global path than the original GDWA, which makes the algorithm improve in execution time, iteration step, and mileage.

3.2 | Experiment of fuzzy controller

We validate the algorithm in the simulation environment shown in Figure 7, and the parameters of the experiment are shown in Table 8. We conducted the experiment 15 times, and the data obtained are shown in Tables 9–11. The significance analysis of the data difference obtained from the experiment is shown in Figure 16.

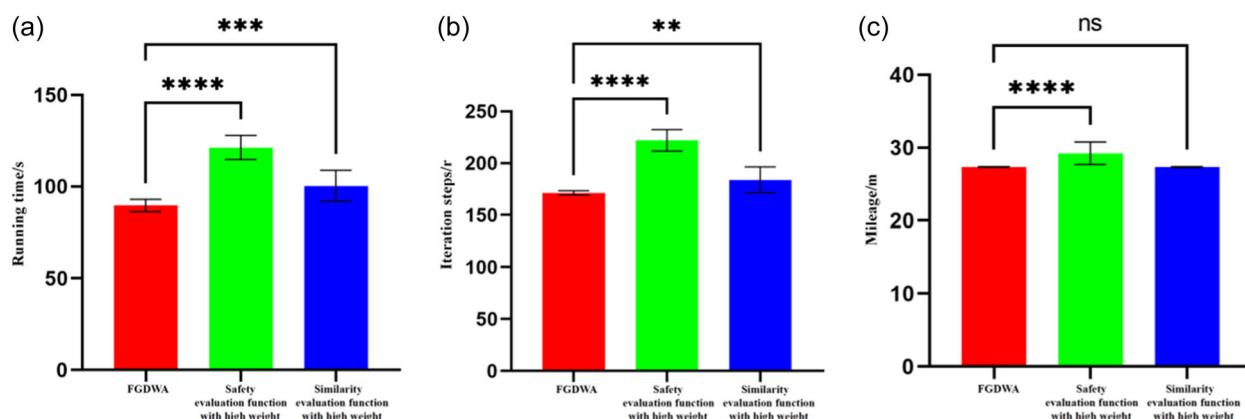


FIGURE 16 Significance analysis of data differences. (a) Significance analysis of running time differences, (b) significance analysis of iteration step differences, and (c) significance analysis of mileage differences. FGDWA, fuzzy-controlled guided dynamic window approach. [Color figure can be viewed at wileyonlinelibrary.com]

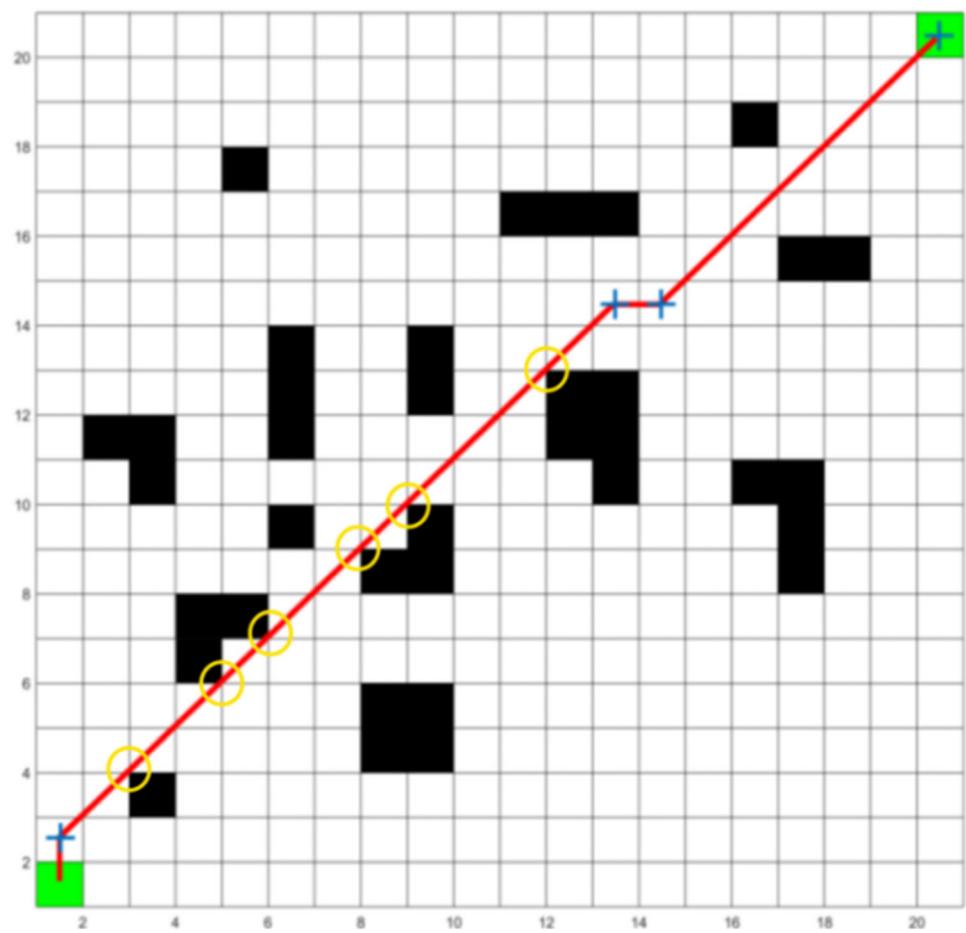


FIGURE 17 Collision-prone locations in the global path. [Color figure can be viewed at wileyonlinelibrary.com]

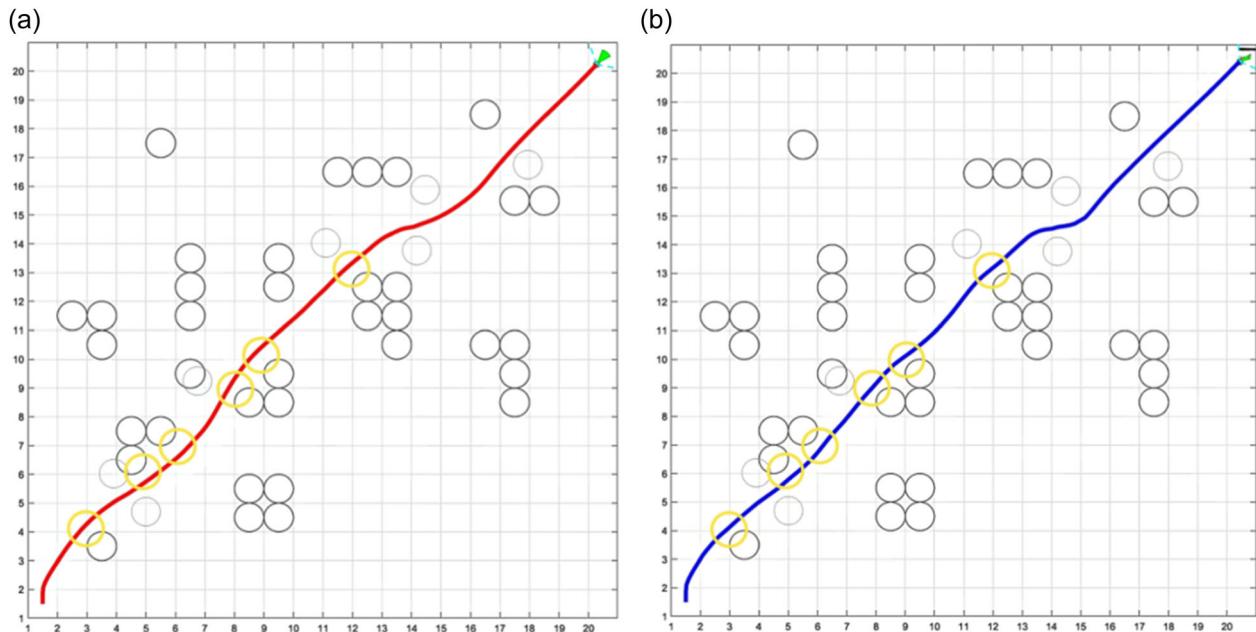


FIGURE 18 Distance comparison of the actual path at danger points. (a) Path of fuzzy-controlled guided dynamic window approach and (b) path of similarity function with high-weight. [Color figure can be viewed at wileyonlinelibrary.com]

From the results, we can see that FGDWA significantly reduced the running time and iteration step compared with the other two fixed-weight GDWAs. In terms of total mileage, FGDWA significantly reduced the total mileage of robot movement compared with the GDWA with a high-weight safety evaluation function. Although there is no obvious difference compared with the GDWA with a high-weight similarity evaluation function, it can be seen in actual paths that the path with a high-weight safety evaluation function is closer to obstacles in some positions than the path obtained by FGDWA. As shown in Figure 17, the red line in the figure is the ideal global path obtained by the global path planning algorithm, in which the positions circled by yellow circles are the dangerous points prone to collision, and Figure 18 is the path obtained by FGDWA and GDWA with high-weight similarity evaluation function in one experiment. It can be seen that the path of FGDWA near the dangerous points is farther from obstacles than the path with high-weight of safety evaluation function. We counted the average distance from the dangerous points of the path to the obstacle obtained by FGDWA and GDWA with high-weight similarity evaluation function in 15 times experiments, as shown in Table 12, and the difference significance analysis is shown in Figure 19. From the data and analysis, we can see that the average distance from the dangerous points obtained by FGDWA is significantly smaller than that of GDWA (with a high-weight similarity evaluation function). To sum up, experiments show that FGDWA, taking

TABLE 12 Average distance between danger points and obstacles.

| Distance from dangerous point to obstacle (m) | FGDWA | Similarity evaluation function with high-weight |
|---|-------|---|
| 1st | 0.48 | 0.32 |
| 2nd | 0.46 | 0.32 |
| 3rd | 0.5 | 0.33 |
| 4th | 0.46 | 0.33 |
| 5th | 0.47 | 0.32 |
| 6th | 0.46 | 0.31 |
| 7th | 0.46 | 0.33 |
| 8th | 0.47 | 0.32 |
| 9th | 0.48 | 0.3 |
| 10th | 0.53 | 0.29 |
| 11th | 0.46 | 0.32 |
| 12th | 0.48 | 0.31 |
| 13th | 0.46 | 0.33 |
| 14th | 0.5 | 0.3 |
| Average | 0.48 | 0.32 |
| Effect (%) | - | 50 |

Abbreviation: FGDWA, fuzzy-controlled guided dynamic window approach.

both the running time and security into consideration, has better adaptability to the actual environment.

Figure 20 shows the changes in safety evaluation function weight β and similarity evaluation function weight δ of adaptive control by fuzzy controller in FGDWA in one round of experiments. From the figure, we can see that the fuzzy controller actively adjusts these two parameters during the robot moving process, which shows that the fuzzy controller designed in our research is feasible and effective.

3.3 | Experiments in complex terrain environment

In this section, we integrated the algorithms for simulation and physical verification. In the actual three-dimensional physical world, mobile robots themselves have many physical constraints, so it is necessary to combine the path planning algorithms with the mobile robot, and conduct simulation experiments in the physical environment to verify the effectiveness of the algorithm.

3.3.1 | Introduction of global path planning algorithm in this research

Compared with single-objective global path planning, multiobjective global path planning has become a hot spot in global path planning

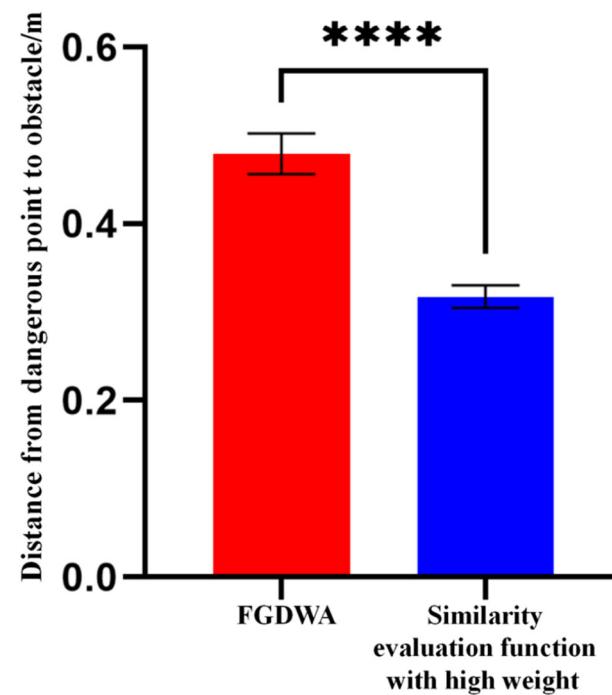


FIGURE 19 Significance analysis of average distance difference between danger points and obstacles. FGDWA, fuzzy-controlled guided dynamic window approach. [Color figure can be viewed at wileyonlinelibrary.com]

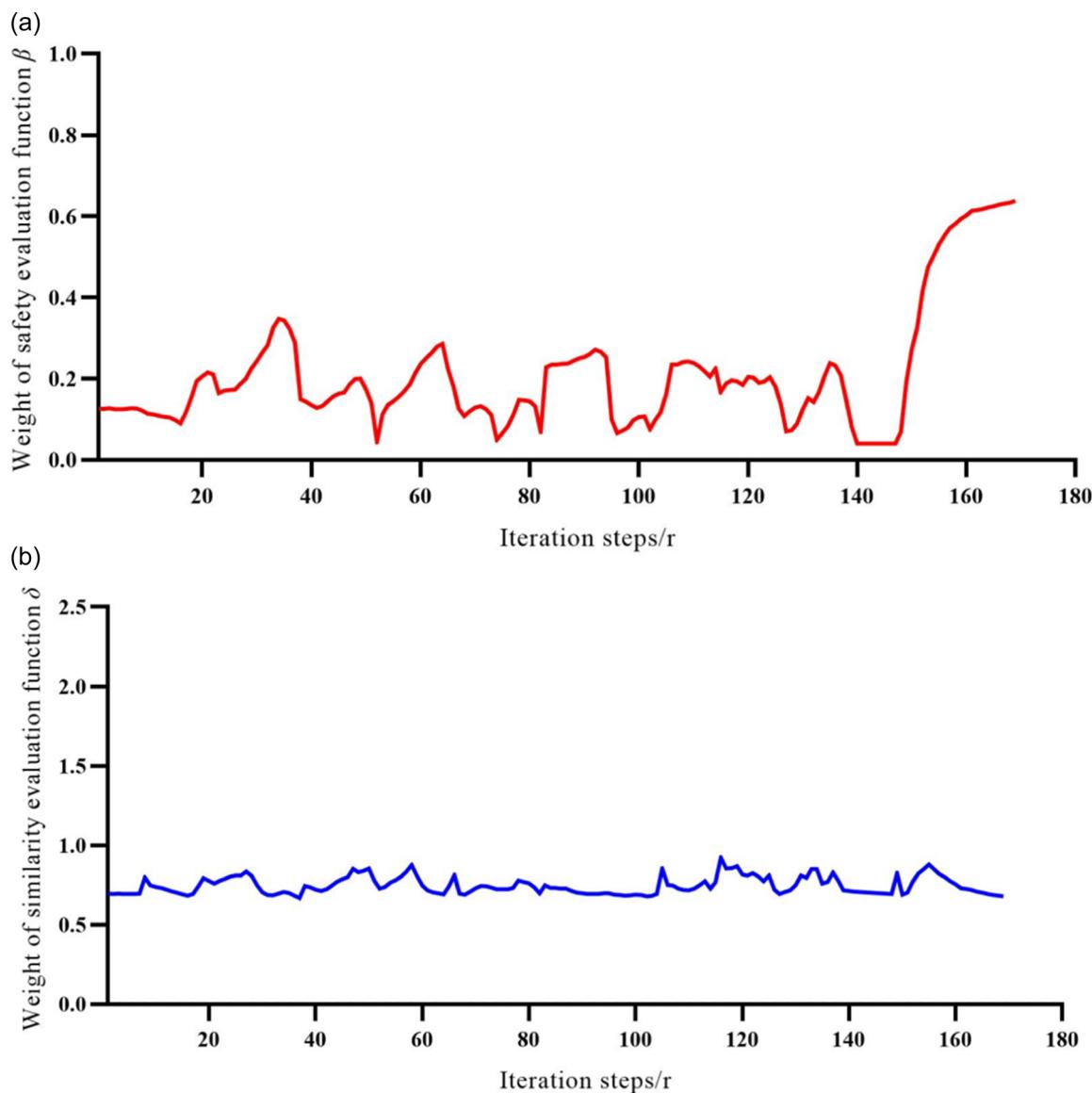


FIGURE 20 Change of fuzzy-controlled guided dynamic window approach weights adapted by fuzzy control in one round. (a) Variation of safety function weight β and (b) variation of similarity function weight δ . [Color figure can be viewed at wileyonlinelibrary.com]

research, and several multiobjective global path planning algorithms have been studied and proposed by researchers, such as path planning based on the genetic algorithm (Singh & Thongam, 2019), path planning based on the particle-swarm optimization algorithm (Ajeil, Ibraheem, Sahib, et al., 2020; Thabit & Mohades, 2019; Wang et al., 2018), and path planning based on multiobjective ant colony optimization algorithm (Ajeil, Ibraheem, Azar, et al., 2020; Ali et al., 2020). The above multiobjective global path planning algorithms usually consider path length and path smoothness, and a few of them include path safety in the planning objectives. However, the safety planning only considers the distance between obstacles and the robot, without considering the safety problems caused by terrain. In Liu, Liu et al. (2022), we proposed a multiobjective path planning algorithm for mobile robot based on a genetic algorithm with an elitist strategy and a new optimization objective: path falling risk, which can effectively improve the safety of mobile robots when passing through complex terrain environments.

3.3.2 | Integration of algorithms

To facilitate the invocation of algorithms, we integrated the global path planning algorithm and the local path planning algorithm into the same Matlab graphical user interface software, as shown in Figure 21. The algorithm running environment is Matlab R2020b, algorithm execution computer configuration is as follows: Core i7 (2.3 GHz), 16 GB RAM, Windows 10.

The integrated algorithm brief flow is as follows:

- (1) Improved A* algorithm for global path planning to obtain global path.
- (2) Processing global path information, including extracting key nodes and calculating the equations of each segment of the global path.
- (3) FGDWA obtains the optimal speed and controls the robot's movement until it reaches the endpoint.

Figure 22 shows the flowchart of the integrated algorithm.

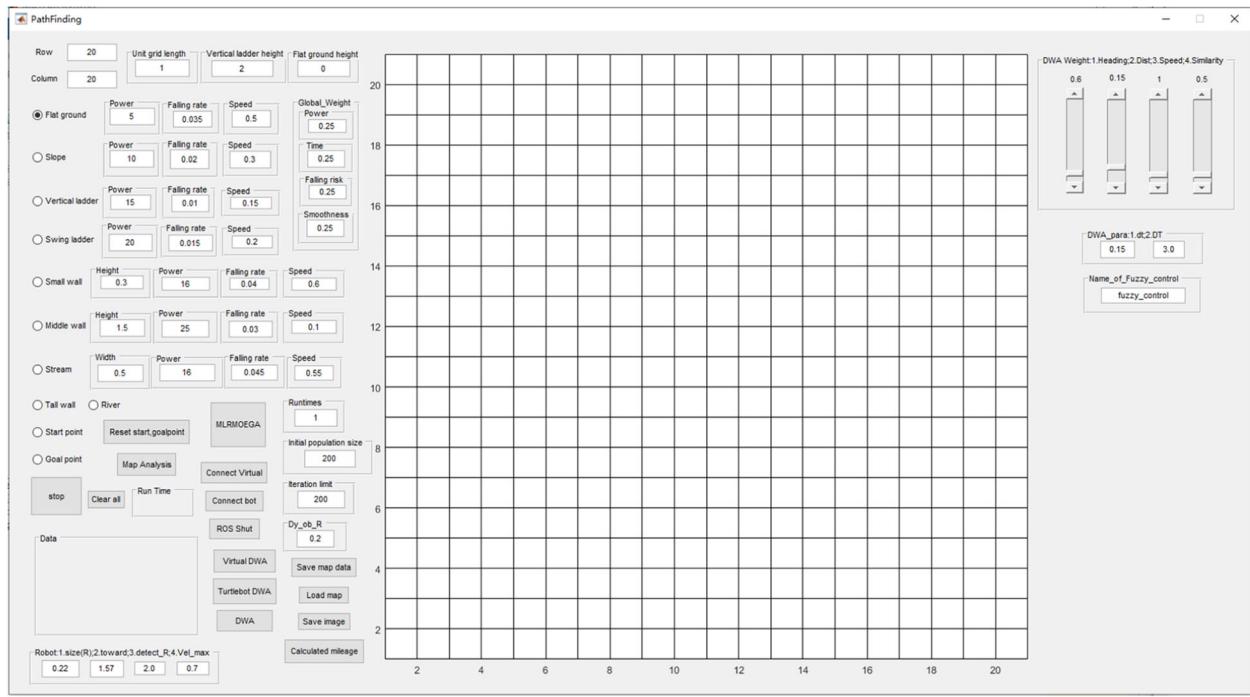


FIGURE 21 Algorithm software based on Matlab graphical user interface.

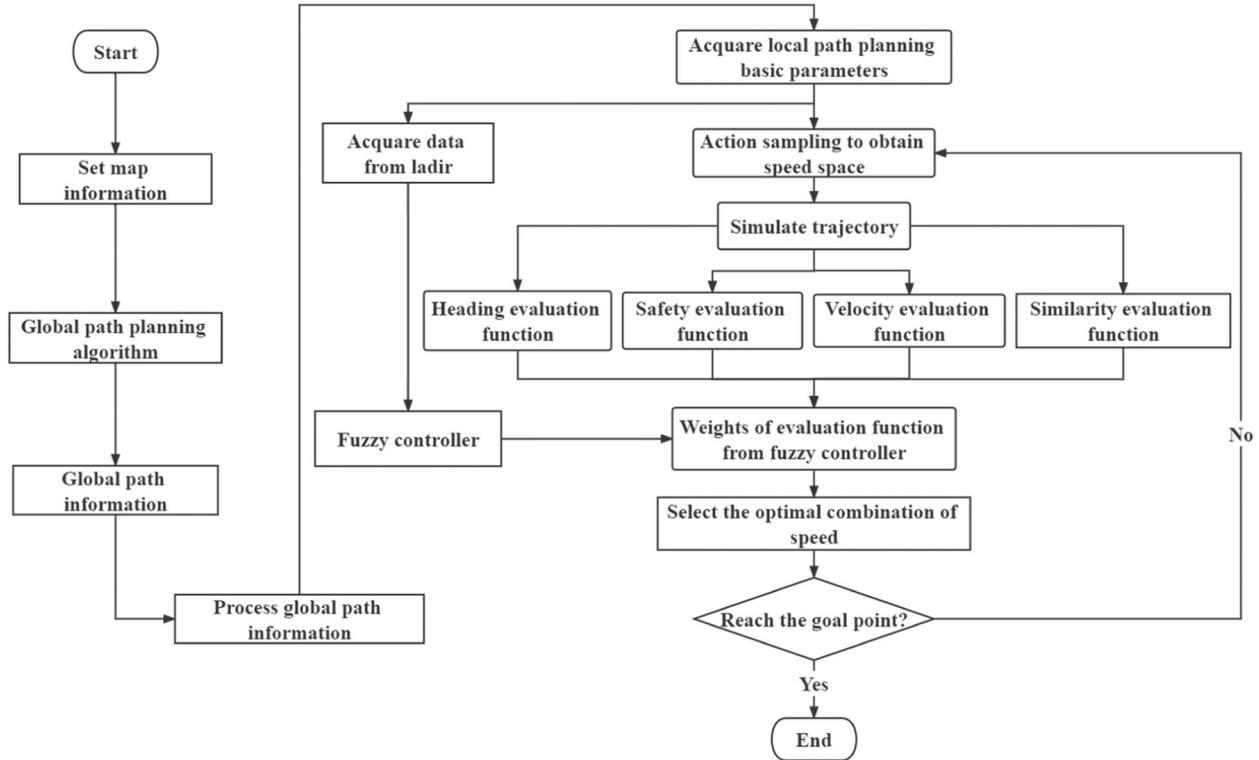


FIGURE 22 Flowchart of the integration algorithm.

3.3.3 | Gazebo simulated environment experiment

Figure 23 is a simulation of a complex terrain environment built in the modeling software Blender. The environment contains a variety of

terrains, such as elevated platforms, steep slopes, slopes, deep pits, and dense obstacle areas. The abstracted grid map is shown in Figure 24.

In Figure 24, we use colored grids to represent the kinds of slope terrain with different degrees. These slopes constitute the

multiple terrain shown in Figure 23. We divide the slopes into four grades according to the different angles of slopes, and different slopes have different passing indicators, which will affect the robot's

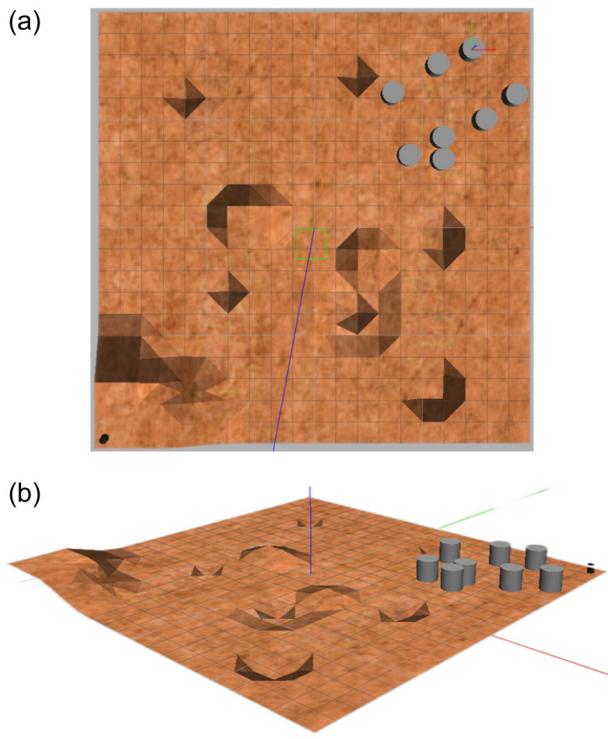


FIGURE 23 Simulation of complex terrain environment. (a) Top view and (b) aerial view. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

moving performance. In the upper right corner of the map in Figure 23, there are multiple obstacles that are not described in the abstract grid map shown in Figure 24. We regard them as dynamic obstacles to test the dynamic obstacle avoidance ability of the proposed algorithm.

3.3.4 | Process of experiment

We use the global path planning algorithm in Liu, Liu et al. (2022) to do global path planning, and then use the proposed algorithm FGDWA and traditional GDWA for local path planning, and then compare the performance of the two algorithms to verify the effectiveness and superiority of the proposed algorithms.

Table 13 shows the parameters of FGDWA, and Table 14 shows the parameters of the GDWA algorithm. According to experimental experience, if the weight of the safety evaluation function is relatively high, the robot is more inclined to bypass obstacles, which causes the robot to deviate from the global path and fall into the local optimal problem. If the weight of the velocity evaluation function is relatively high, the robot tends to follow the simulated trajectory with the maximum linear velocity. When encountering dynamic obstacles and needing to avoid them, the speed of the robot may change suddenly, resulting in poor path smoothness.

Time resolution and Simulation time of per cycle are two parameters that affect the generated of simulated trajectories in DWA. If the time resolution is set too large, the resolution of simulated trajectories will be low, and the width of the dynamic window

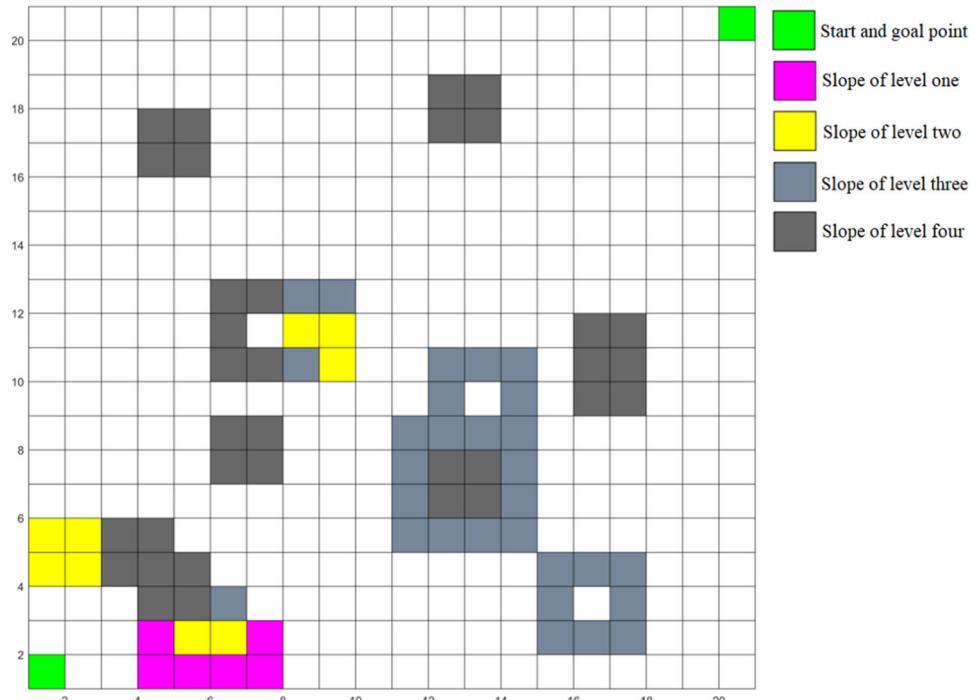


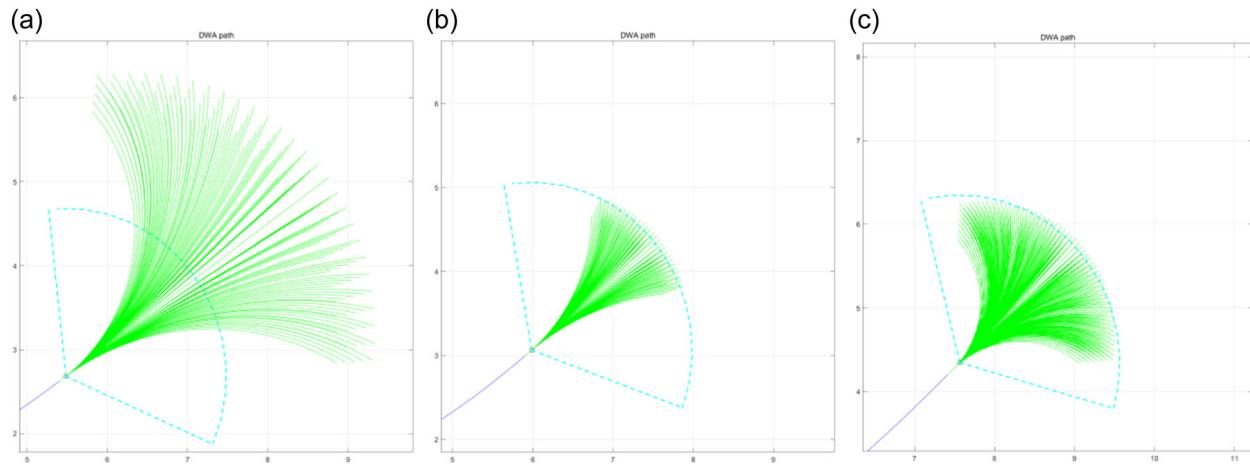
FIGURE 24 Abstracted grid map. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

TABLE 13 Parameters of fuzzy-controlled guided dynamic window approach.

| Time resolution dt (s) | Simulation time per cycle perDT (s) | Weight of heading evaluation function α | Weight of velocity evaluation function γ |
|-----------------------------|--|---|--|
| 0.15 | 3 | 0.5 | 1 |

TABLE 14 Parameters of guided dynamic window approach.

| Time resolution dt (s) | Simulation time per cycle perDT (s) | Weight of heading evaluation function α | Weight of safety evaluation function β | Weight of velocity evaluation function γ |
|--------------------------|--|---|---|--|
| 0.15 | 3 | 0.5 | 0.15 | 1 |

**FIGURE 25** Global path. (a) Time resolution, 0.15 s; simulation time per cycle, 6 s; (b) time resolution, 0.3 s; simulation time per cycle, 3 s; (c) time resolution, 0.15 s; simulation time per cycle, 3 s. [Color figure can be viewed at wileyonlinelibrary.com]

will be narrower. If the simulation time is set too large, the simulated trajectories will be too long, and it not only wastes computing resources but also causes a large deviation between the simulated trajectory and the trajectory that the mobile robot can move in each computing cycle of the algorithm. As shown in Figure 25c, the simulated trajectories generated by the algorithm under these two parameters we set are consistent with the range of the detection window.

Figure 26 shows the global path obtained by global path planning. The blue + indicates the key nodes of the global path.

The experimental process is as follows:

- (1) Construct a grid map according to a simulated complex terrain environment.
- (2) Use the global path planning algorithm to generate the global path.
- (3) Local path planning algorithm to control robot movement and record the actual path of local path planning.
- (4) Record the execution time, iteration step, and mileage of the local path planning algorithm.

3.3.5 | Results of the experiment

We conducted 15 times local path planning experiments in a simulated complex terrain environment. Figure 27 shows one of the actual paths of FGDWA and GDWA in several experiments. The execution time, iteration step, and mileage of 15 times experiments are shown in Tables 15–18.

As shown in Figure 27a, under the effect of the proposed similarity evaluation function, the actual path of FGDWA is very close to the global path, and it is far away from some dangerous locations in the complex terrain environment.

In 15 times experiments, since the actual path of GDWA is not constrained by the similarity evaluation function, and the weight of safety evaluation function β cannot adjust in real time according to the actual situation of the surrounding obstacles, the robot is trapped by "C"-shaped obstacles combination and collision with obstacles for several times as shown in Figures 27b and 28. For under the constrained of the similarity evaluation function, FGDWA actual paths are closer to the global optimal path than GDWA actual paths, in the case that successfully reaching the goal point, the execution time,

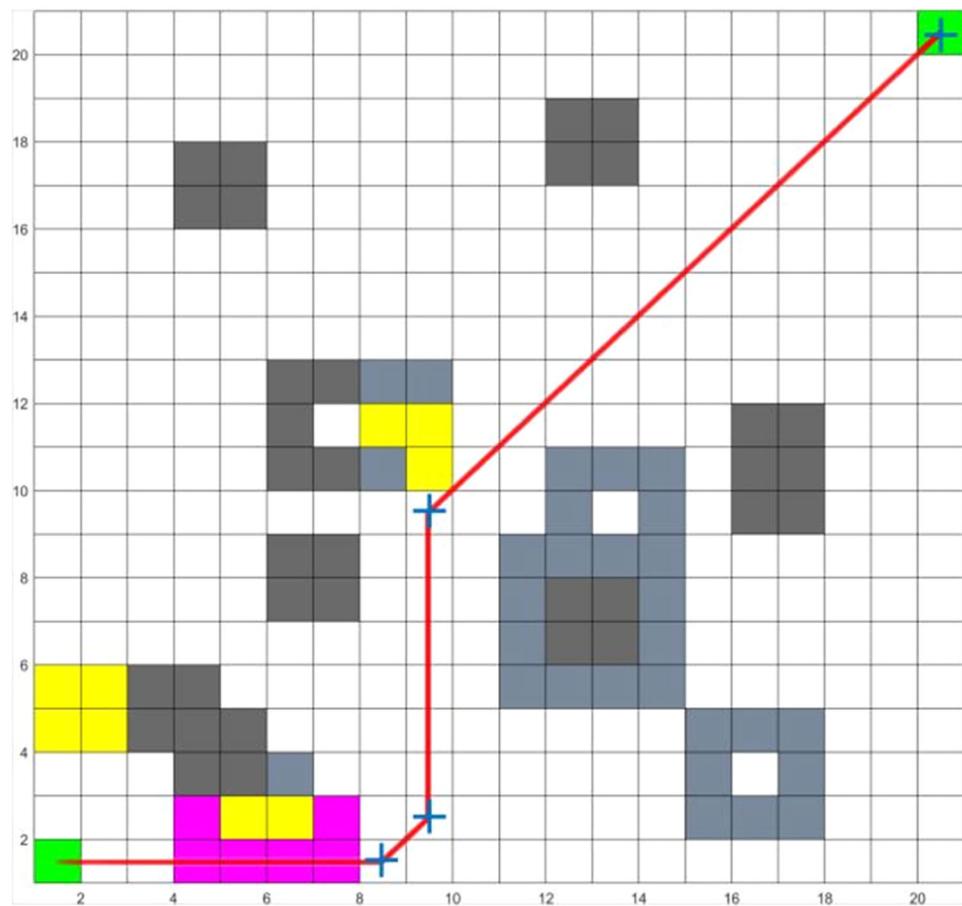


FIGURE 26 Global path. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

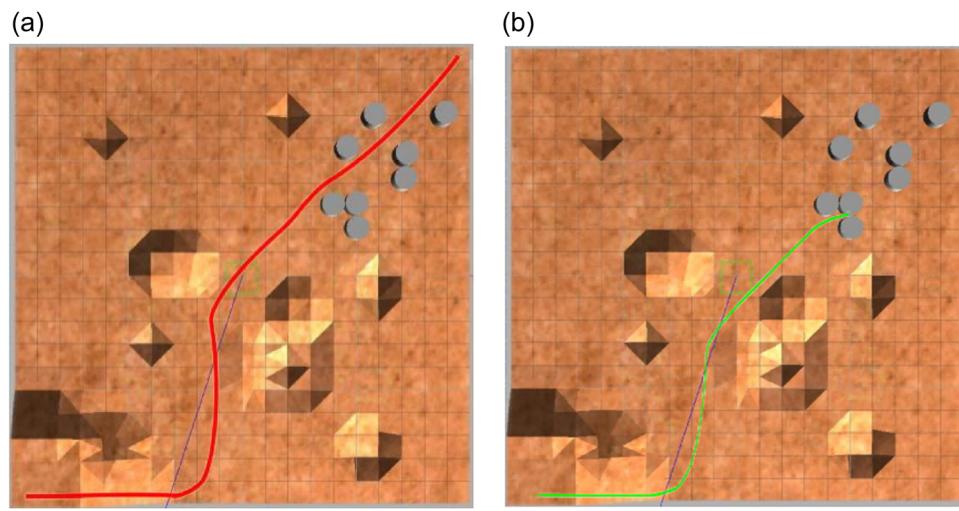


FIGURE 27 Actual paths of local path planning. (a) Actual path generated by fuzzy-controlled guided dynamic window approach and (b) actual path generated by guided dynamic window approach. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

iteration step, and mileage of GDWA actual paths are not as good as FGDWA actual paths.

The significance analysis of the data difference obtained from the experiment is shown in Figure 29. We remove the unfinished

experiments of GDWA in data analysis. From Figure 29 we can see that there are significant differences between the three sets of data of FGDWA path and GDWA path, and the FGDWA path is superior to the GDWA path.

TABLE 15 Running time of 15 experiments.

| Running time (s) | FGDWA | GDWA |
|------------------|-------|------------|
| 1st | 92.2 | Unfinished |
| 2nd | 88.6 | 113.4 |
| 3rd | 88.9 | 113.6 |
| 4th | 90.2 | 115.8 |
| 5th | 81.9 | Unfinished |
| 6th | 87.6 | 113.5 |
| 7th | 88.6 | 113.2 |
| 8th | 88.7 | 112.4 |
| 9th | 88.6 | 112.3 |
| 10th | 89.7 | 112.4 |
| 11th | 88.9 | 111.4 |
| 12th | 93.5 | Unfinished |
| 13th | 90.3 | 112.6 |
| 14th | 89.2 | 114.1 |
| 15th | 90.1 | 113.6 |
| Average | 89.1 | - |

Abbreviations: FGDWA, fuzzy-controlled guided dynamic window approach; GDWA, guided dynamic window approach.

TABLE 16 Iteration step of the 15 experiments.

| Iteration step (r) | FGDWA | GDWA |
|--------------------|-------|------------|
| 1st | 185 | Unfinished |
| 2nd | 171 | 224 |
| 3rd | 181 | 222 |
| 4th | 182 | 225 |
| 5th | 171 | Unfinished |
| 6th | 180 | 221 |
| 7th | 181 | 220 |
| 8th | 183 | 219 |
| 9th | 180 | 222 |
| 10th | 184 | 222 |
| 11th | 186 | 219 |
| 12th | 185 | Unfinished |
| 13th | 183 | 220 |
| 14th | 186 | 226 |
| 15th | 188 | 224 |
| Average | 182 | - |

Abbreviations: FGDWA, fuzzy-controlled guided dynamic window approach; GDWA, guided dynamic window approach.

TABLE 17 Mileage of 15 experiments.

| Mileage (m) | FGDWA | GDWA |
|-------------|-------|------------|
| 1st | 30.76 | Unfinished |
| 2nd | 30.82 | 30.84 |
| 3rd | 30.80 | 30.90 |
| 4th | 30.75 | 30.83 |
| 5th | 30.84 | Unfinished |
| 6th | 30.82 | 30.90 |
| 7th | 30.85 | 30.89 |
| 8th | 30.81 | 30.87 |
| 9th | 30.84 | 30.83 |
| 10th | 30.76 | 30.86 |
| 11th | 30.85 | 30.93 |
| 12th | 30.83 | Unfinished |
| 13th | 30.70 | 30.84 |
| 14th | 30.80 | 30.87 |
| 15th | 30.78 | 30.85 |
| Average | 30.80 | - |

Abbreviations: FGDWA, fuzzy-controlled guided dynamic window approach; GDWA, guided dynamic window approach.

TABLE 18 Finishing rate of experiments.

| | FGDWA | GDWA |
|--------------------|-------|------|
| Finishing rate (%) | 100 | 80 |
| Effect (%) | 25 | - |

Abbreviations: FGDWA, fuzzy-controlled guided dynamic window approach; GDWA, guided dynamic window approach.

3.4 | Conclusion of experiments

In this section, we verify the effectiveness of the proposed similarity evaluation function. From the results, we can see that the proposed similarity evaluation function effectively improves the guiding effect of the global path and effectively avoids the problem of the local path planning algorithm falling into local optimal.

Then, we verified the effectiveness of the proposed fuzzy controller. It can be seen that the fuzzy controller can adjust the weight value of the safety evaluation function and similarity evaluation function in real time, so that FGDWA can effectively avoid collision when it moves following the global path.

Finally, we compared FGDWA with GDWA in a simulated complex terrain environment, and it can be seen that the FGDWA is superior to the GDWA in execution time, iteration step, and mileage.

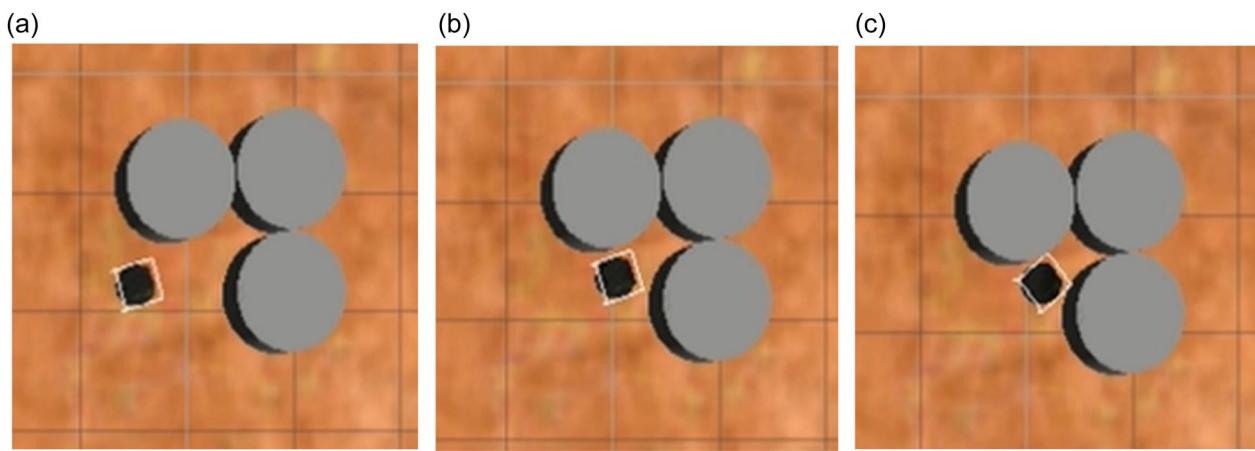


FIGURE 28 Robot trapped by “C”-shaped obstacles combination and collision with obstacles in guided dynamic window approach local path. (a) Robot approaches “C”-shaped obstacles combination, (b) Robot enters “C”-shaped obstacles combination and (c) Robot trapped by “C”-shaped obstacles combination. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

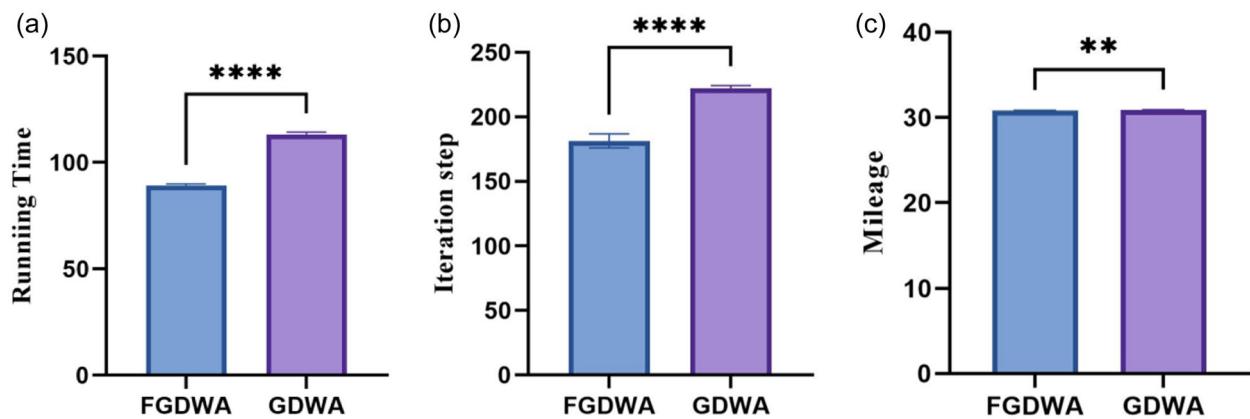


FIGURE 29 Significance analysis of data differences. FGDWA, fuzzy-controlled guided dynamic window approach; GDWA, guided dynamic window approach. (a) Significance analysis of data differences of Running time, (b) Significance analysis of data differences of Iteration step, and (c) Significance analysis of data differences of Mileage. [Color figure can be viewed at [wileyonlinelibrary.com](#)]

4 | CONCLUSIONS

The DWA algorithm is easy to fall into local optimal without global path guidance or under the guidance of temporary target points of the global path. In complex environments, if the local path deviates from the global path, the time cost may be increased, and the robot may enter dangerous areas or even be damaged. To improve these problems, a new similarity evaluation function is proposed for the DWA algorithm, and the DTW algorithm is used to evaluate the similarity of trajectories.

Then we test the performance of the proposed similarity evaluation function and compare with traditional DWA algorithm, the result shows that the similarity evaluation function effectively improves the coincidence between local path and global path and reduces algorithm execution time by 0.7% and mileage by 2.1%.

However, under the fixed evaluation function weights, the DWA algorithm has poor adaptability to the environment and cannot adjust

the weights of evaluation functions in real time according to the actual situation. Therefore, a fuzzy controller is designed to improve the adaptability of the DWA algorithm to the actual environment. We design two input parameters: obstacle density I and path deviation D , to evaluate the actual environment around the mobile robot, then adjust the DWA algorithm's bias to similarity and safety through a fuzzy controller.

We test the performance of the proposed similarity evaluation function and the proposed fuzzy controller in a simple environment. Then, we compared the performance of FGDWA and GDWA in a simulated complex terrain environment. According to the data from the experiments, the fuzzy controller reduces algorithm execution time by 10.8% and improves the average distance between the mobile robot and the obstacle at the global path's danger points by 50%, and in simulated complex terrain environment, the finishing rate of experiments improves by 25%.

In summary, compared with the traditional GDWA, the actual path generated by our improved algorithm FGDWA is closer to the

global path, which has improved in running time, iteration step, and mileage. The phenomenon is that traditional DWA is prone to fall into local optimal and the adaptability of the algorithm to different environments is also improved.

5 | FURTHER RESEARCH

In this paper, the three-dimensional environment is idealized into two-dimensional maps, and the local path planning in our research is only the control of the speed in two dimensions. Path planning algorithm is no longer limited to the two-dimensional environment nowadays, the robot industry is developing rapidly, such as four-rotor UAV, so that the robot working environment is gradually complex and diversified, and the research prospect of this algorithm should be do path planning in three-dimensional environment. Therefore, this research will focus on expanding this algorithm into three-dimensional space in the future.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

ORCID

Aizun Liu  <https://orcid.org/0000-0002-4448-3521>
Chong Liu  <https://orcid.org/0000-0001-9352-4994>

REFERENCES

- Agrawal, R., Imieliński, T. & Swami, A. (2005) *Mining association rules between sets of items in large databases*. New York, NY, United States: ACM SIGMOD Record, pp. 207–216. Available from: <https://doi.org/10.1145/170036.170072>
- Ajeil, F.H., Ibraheem, I.K., Azar, A.T. & Humaidi, A.J. (2020). Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. *Sensors*, 20, 1880.
- Ajeil, F.H., Ibraheem, I.K., Sahib, M.A. & Humaidi, A.J. (2020) Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. *Applied Soft Computing*, 89, 106076.
- Ali, H., Gong, D., Wang, M. & Dai, X. (2020) Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in grid-based environment. *Frontiers in Neurorobotics*, 14, 44.
- Bai, X., Jiang, H., Cui, J., Lu, K., Chen, P. & Zhang, M. (2021) UAV path planning based on improved A* and DWA algorithms. *International Journal of Aerospace Engineering*, 2021, 1–12. Available from: <https://doi.org/10.1155/2021/4511252>
- Bao, J., Li, R., Yi, X. & Zheng, Y. (2016) Managing massive trajectories on the cloud. In: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, United States: Association for Computing Machinery, pp. 1–10. Available from: <https://doi.org/10.1145/2996913.2996916>
- Cai, J., Wan, M., Huang, Z. & Liu, Z. (2022) An improved DWA path planning algorithm integrating global JPS strategy. In: *2022 2nd International Conference on Computer, Control and Robotics (ICCCR)* Shanghai, China: IEEE, pp. 20–26. Available from: <https://doi.org/10.1109/ICCCR54399.2022.9790216>
- Campion, G. & Chung, W. (2008) Wheeled robots. In: Siciliano, B. & Khatib, O. (Eds.) *Springer handbook of robotics*. Berlin, Heidelberg: Springer. Available from: https://doi.org/10.1007/978-3-540-30301-5_18
- Castillo, O., Trujillo, L. & Melin, P. (2006) Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing*, 11, 269–279. Available from: <https://doi.org/10.1007/s00500-006-0068-4>
- Chen, L. & Ng, R. (2009) On the marriage of L_p -norms and edit distance. In: *Proceedings of the 2004 VLDB Conference*, vol. 30. St. Louis, MO, United States: Morgan Kaufmann, pp. 792–803. Available from: <https://doi.org/10.1016/b978-012088469-8.50070-x>
- Du, C., Lee, K.-H. & Newman, W. (2015) Manipulation planning for the Atlas humanoid robot. In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. Presented at the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO), Bali, Indonesia. Bali, Indonesia: IEEE. Available from: <https://doi.org/10.1109/robio.2014.7090482>
- Hao, W., Haiming, L. & Jiancheng, J. (2019) Handling forecasting problems based on fuzzy time series model and model error learning. *Applied Soft Computing Journal*, 78, 109–118.
- Hu, Y., Li, J. & Jiang, L. (2023) Local path planning combined with the motion state of dynamic obstacles. In: Pan, L., Zhao, D., Li, L. & Lin, J. (Eds.) *Bio-Inspired Computing: Theories and Applications. BIC-TA 2022. Communications in Computer and Information Science*, vol. 1801. Singapore, Singapore: Springer. Available from: https://doi.org/10.1007/978-981-99-1549-1_46
- Keogh, E. & Ratanamahatana, C.A. (2005) Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7, 358–386. Available from: <https://doi.org/10.1007/s10115-004-0154-9>
- Klaassen, B. & Paap, K.L. (2003) GMD-SNAKE2: a snake-like robot driven by wheels and a method for motion control. In: *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 4. Detroit, MI, United States: IEEE, pp. 3014–3019. Available from: <https://doi.org/10.1109/robot.1999.774055>
- Li, P., Hao, L., Zhao, Y. & Lu, J. (2024) Robot obstacle avoidance optimization by A* and DWA fusion algorithm. *PLoS ONE*, 19(4), e0302026. Available from: <https://doi.org/10.1371/journal.pone.0302026>
- Lin, Z., Yue, M., Chen, G. & Sun, J. (2021) Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles. *Transactions of the Institute of Measurement and Control*, 44(1), 121–132. Available from: <https://doi.org/10.1177/01423312211024798>
- Liu, C., Liu, A., Wang, R., Zhao, H. & Lu, Z. (2022) Path planning algorithm for multi-locomotion robot based on multi-objective genetic algorithm with elitist strategy. *Micromachines*, 13, 616. Available from: <https://doi.org/10.3390/mi13040616>
- Liu, X., Jiang, D., Tao, B., Jiang, G., Sun, Y., Kong, J. et al. (2022) Genetic algorithm-based trajectory optimization for digital twin robots. *Frontiers in Bioengineering and Biotechnology*, 9, 793782. Available from: <https://doi.org/10.3389/fbioe.2021.793782>
- Mai, X., Li, D., Ouyang, J. & Luo, Y. (2021) An improved dynamic window approach for local trajectory planning in the environment with dense objects. *Journal of Physics: Conference Series*, 1884, 012003 (8pp.). Available from: <https://doi.org/10.1088/1742-6596/1884/1/012003>
- Maroti, A., Szaloki, D., Kiss, D. & Tevesz, G. (2013) Investigation of dynamic window based navigation algorithms on a real robot. In: *2013 IEEE 11th International Symposium on Applied Machine Intelligence and Informatics (SAMI 2013)*, vol. 2013. Herľany, Slovakia: IEEE, pp. 95–100. Available from: <https://doi.org/10.1109/SAMI.2013.6480952>
- Orozco-Rosas, U., Montiel, O. & Sepúlveda, R. (2019) Mobile robot path planning using membrane evolutionary artificial potential field. *Applied Soft Computing*, 77, 236–251. Available from: <https://doi.org/10.1016/j.asoc.2019.01.036>

- Orozco-Rosas, U., Picos, K. & Montiel, O. (2019) Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots. *IEEE Access*, 7, 156787–156803. Available from: <https://doi.org/10.1109/access.2019.2949835>
- Orozco-Rosas, U., Picos, K., Pantrigo, J.J., Montemayor, A.S. & Cuesta-Infante, A. (2022) Mobile robot path planning using a QAPF learning algorithm for known and unknown environments. *IEEE Access*, 10, 84648–84663. Available from: <https://doi.org/10.1109/access.2022.3197628>
- Patle, B.K., Babu, L.G., Pandey, A., Parhi, D.R.K. & Jagadeesh, A. (2019) A review: on path planning strategies for navigation of mobile robot. *Defence Technology*, 15, 582–606. Available from: <https://doi.org/10.1016/j.dt.2019.04.011>
- Raiert, M., Blankepoor, K., Nelson, G. & Playter, R. (2010) BigDog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41(2), 10822–10825. Available from: <https://doi.org/10.3182/20080706-5-kr-1001.01833>
- Rubio, F., Valero, F. & Llopis-Albert, C. (2019) A review of mobile robots: concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2), 172988141983959. Available from: <https://doi.org/10.1177/1729881419839596>
- Singh, N.H. & Thongam, K. (2019) Mobile robot navigation using fuzzy-GA approaches along with three path concept. *The Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 43, 277–294.
- Song, B., Tang, S. & Li, Y. (2024) A new path planning strategy integrating improved ACO and DWA algorithms for mobile robots in dynamic environments. *Mathematical Biosciences and Engineering*, 21(2), 2189–2211. Available from: <https://doi.org/10.3934/mbe.2024096>
- Tanaka, K. & Sugeno, M. (1992) Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems*, 45(2), 135–156.
- Thabit, S. & Mohades, A. (2019) Multi-robot path planning based on multi-objective particle swarm optimization. *IEEE Access*, 7, 2138–2147.
- Trautman, P., Ma, J., Murray, R.M. & Krause, A. (2013) Robot navigation in dense human crowds: the case for cooperation. In: *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, pp. 2153–2160. Available from: <https://doi.org/10.1109/ICRA.2013.6630866>
- Wang, B., Li, S., Guo, J. & Chen, Q. (2018) Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. *Neurocomputing*, 282, 42–51.
- Wu, B., Chi, X., Zhao, C., Zhang, W., Lu, Y. & Jiang, D. (2022) Dynamic path planning for forklift AGV based on smoothing A* and improved DWA hybrid algorithm. *Sensors*, 22(18), 7079. Available from: <https://doi.org/10.3390/s22187079>
- Xu, Z., Zhao, A., Zhai, B., Wang, A. & Zhang, L. (2020) Comparative studies of robot navigation. In: Liu, Y., Wang, L., Zhao, L. & Yu, Z. (Eds.) *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery. ICNC-FSKD 2019. Advances in Intelligent Systems and Computing*, vol. 1075. Cham, Cham, Switzerland: Springer. Available from: https://doi.org/10.1007/978-3-030-32591-6_32
- Yan, X., Ding, R., Luo, Q., Ju, C. & Wu, D. (2022) A dynamic path planning algorithm based on the improved DWA algorithm. In: *2022 Global Reliability and Prognostics and Health Management (PHM-Yantai)*. Yantai, China: IEEE, pp. 1–7. Available from: <https://doi.org/10.1109/PHM-Yantai55411.2022.9942106>
- Yin, X., Cai, P., Zhao, K., Zhang, Y., Zhou, Q. & Yao, D. (2023) Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms. *Sensors*, 23(8), 4102. Available from: <https://doi.org/10.3390/s23084102>
- Zhang, C., Yang, X., Zhou, R. & Guo, Z. (2024) A path planning method based on improved A* and fuzzy control DWA of underground mine vehicles. *Applied Sciences*, 14(7), 3103. Available from: <https://doi.org/10.3390/app14073103>
- Zhou, Z., Geng, C., Qi, B., Meng, A. & Xiao, J. (2023) Research and experiment on global path planning for indoor AGV via improved ACO and fuzzy DWA. *Mathematical Biosciences and Engineering*, 20(11), 19152–19173. Available from: <https://doi.org/10.3934/mbe.2023846>

How to cite this article: Liu, A., Liu, C., Li, L., Wang, R. & Lu, Z. (2024) An improved fuzzy-controlled local path planning algorithm based on dynamic window approach. *Journal of Field Robotics*, 1–25.
<https://doi.org/10.1002/rob.22419>