

Research Article

Mobile Robot Path Planning Based on Enhanced Dynamic Window Approach and Improved A* Algorithm

Hongxia Yang  and Xingqiang Teng 

College of Business, Yantai Nanshan University, Shandong, Longkou 265701, China

Correspondence should be addressed to Hongxia Yang; yanghongxia1@nanshan.edu.cn

Received 28 January 2022; Accepted 5 March 2022; Published 22 March 2022

Academic Editor: Shan Zhong

Copyright © 2022 Hongxia Yang and Xingqiang Teng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Path planning is one of the most popular researches on mobile robots, and it is the key technology to realize autonomous navigation of robots. Aiming at the problem that the mobile robot may collide or fail along the planned path in an environment with random obstacles, a robot path planning scheme that combines the improved A* algorithm with an enhanced dynamic window method is proposed. In the improved A* algorithm, in order to improve the algorithm efficiency, so that a single planning path can pass through multiple target points, the search point selection strategy and evaluation function are optimized. In order to achieve local obstacle avoidance and pursuit of dynamic target points in dynamic and complex environments, an online path planning method combining enhanced dynamic window algorithm and global path planning information is proposed. The preview deviation angle tracking method is used to successfully capture moving target points. It also improves the efficiency of path planning and ensures that on the basis of the global optimal path, the random obstacle can be avoided in real time so that the robot can reach the target point smoothly. The simulation results show that compared with other methods, the proposed method achieves excellent global and local path planning performance, the planned trajectory is smoother, and the search efficiency is higher in complex environments.

1. Introduction

Mobile robot is an important branch of robotics research [1]. With the continuous development of the world economy and technology, mobile robots appear more and more frequently in scientific research, production environment, and daily life [2]. At the same time, with the development of computers and control technology, the application fields of mobile robots are becoming more and more extensive [3].

Mobile robot needs to plan a path from the initial position to the target position in the work scene. The path should meet a series of requirements such as short length, high efficiency, and high safety, and it must be able to avoid static and dynamic obstacles along the way [4]. At the same time, mobile robots should have certain computing capabilities to calculate the shortest and safest route in real time to save time and reserve energy [5].

According to the characteristics of mobile robot path planning, it can be divided into the intelligent search algorithm, artificial intelligence algorithm, geometric model algorithm, and local obstacle avoidance algorithm [6]. The intelligent search algorithm uses randomly generated initial solutions or sampling points to approach the optimal solution through multiple iterations. The biggest characteristic of this type of algorithm is randomness, so its solution is not unique [7]. Algorithms based on artificial intelligence include methods such as Q learning and deep learning, but such methods require a large number of different types of training samples, which limit their practicality in the real world [8].

The geometric model-based path planning method is to construct a geometric model on the basis of known environment, then select an appropriate path, and adjust the feasible solution based on the optimal strategy in real-time [9]. The paths obtained by such methods are all nonsmooth

paths, so optimization is needed to achieve the smooth turning of the mobile robot. Such methods include A* algorithm, Voronoi graph, and so on. In the global path planning, the A* algorithm will select the square with the lowest current cost value until the search reaches the end, thereby planning the path with the lowest total costs. However, the traditional A* algorithm suffers from poor global optimization capability, and there are too many redundant points and inflection points in the planned path [10].

The local obstacle avoidance algorithm is used to enhance the obstacle avoidance ability of mobile robots and improve safety. It aims to move the robot away from obstacles and plan a safe collision-free path. Commonly used local obstacle avoidance algorithms include artificial potential field method, dynamic window approach (DWA), and so on [11]. The DWA is a method of sampling the surroundings at the current moment to obtain the robot's action state at the next moment. This method can quickly reach the target point while avoiding collisions between the robot and obstacles in the search space. However, it is highly dependent on global parameters and is prone to failure in unknown environments [12].

In order to solve the above problems, this paper proposes a random obstacle avoidance method for mobile robots that combines the improved A* algorithm with an enhanced dynamic window approach (EDWA), which takes into account the smoothness of the planned path and the search efficiency of the A* algorithm. Then based on the global planning path information and dynamic environment changes, the EDWA is used to complete the local path planning, so as to make up for the shortcomings of the traditional A* algorithm with poor timeliness. The proposed method pursues dynamic obstacles on the basis of global path planning information, thereby planning the optimal path and improving real-time performance and flexibility.

The rest of this paper is organized as follows: the second part is the related work, which introduces the existing advanced robot path planning methods. The third part is the improved A* algorithm. The fourth part is the enhanced version of the dynamic window method. The fifth part is the experimental part. The sixth part is the conclusion.

2. Related Methods

The A* algorithm has the advantages of short planned paths and fast calculation speed, and it is a widely used global path planning method. Yuan and Han [13] successfully implement the improved A* algorithm to achieve robot path planning in a static environment. Yu et al. [14] use an improved Dijkstra algorithm to complete global path planning, but its local path search speed is very slow, and a smooth and safe optimal path cannot be quickly planned. Kaplan et al. [15] introduce an improved multiorder B-spline curve for path smoothing, but this method cannot complete the smoothing of the dynamic path, and the running process is cumbersome. The efficiency of the algorithm is poor. When the environment changes or the robot is affected by other external factors, the established path will fail.

Artificial potential field method and DWA are currently widely used local path planning methods. Aiming at the problem that the robot is regarded as a single point in most planning, which leads to the problem of being unable to pass the narrowband, Kiss and Tevesz [16] propose a global dynamic window navigation scheme based on model predictive control in which the weighted objective function is omitted. Aiming at the energy consumption problem in the path planning process, Henkel et al. [17] propose an efficient local path planner suitable for omnidirectional mobile robot navigation in a dynamic environment. The experimental results show that compared with the DWA algorithm, the energy consumption is reduced by 9.79%.

In practical applications, robots often encounter unexpected obstacles such as unknown obstacles during the planning process, which may cause the robots to collide. Therefore, many methods combine the strong obstacle avoidance ability of local path planning algorithms with global path planning methods to improve performance. Aiming at the problem that the classic A* algorithm cannot be applied in a dynamic environment, Zhong et al. [18] propose a hybrid method combining improved A* algorithm and DWA algorithm. Experimental results show that the path planned by the improved algorithm is smoother and more efficient. But the disadvantage of this method is that it is only applicable to target points in the same direction. Wu et al. [19] propose a hybrid algorithm combining beetle antenna search algorithm and artificial potential field algorithm for the dynamic path planning problem. The experimental results verify the effectiveness and superiority of the method. The method proposed by Kashyap et al. [20] combines the DWA algorithm and teaching and learning optimization. Path planning and obstacle avoidance can be successfully implemented in both the dynamic terrain and the dynamic terrain.

For the path planning problem of mobile robots in complex environments, Ma et al. [21] propose a method combining improved A* algorithm, minimum snap trajectory generation, and the timed elastic band, which improves the efficiency and flexibility of mobile robot path planning. The experimental results show that this method can shorten the length of the planned path, reduce the total turning angle, and effectively complete the global optimal path planning and local real-time obstacle avoidance.

3. Improved A* Algorithm

3.1. Traditional A Algorithm.* The A* algorithm is a heuristic search algorithm that obtains the planned path of the mobile robot by continuously searching for the path approaching the destination. This method is simple and fast, and the heuristic search is very targeted. It only needs to search a part of the state space of the problem, so it can achieve the purpose of narrowing the search range and reducing the complexity of the problem, and the path finding efficiency is very high.

The A* algorithm can be expressed as follows:

$$f(n) = g(n) + h(n), \quad (1)$$

where n is the current node, $f(n)$ is the cost value of the optimal path from the starting node to the target point, and $g(n)$ is the depth factor, which is the cost of the optimal path from the starting node to node n found by the heuristic search algorithm. $h(n)$ is an estimated cost value of the optimal path from node n to the target node. The value of each decision is evaluated according to the evaluation function, and it is decided which scheme to try first in order to obtain the shortest path.

3.2. Improved A* Algorithm. The traditional A* algorithm has the following two shortcomings: (1) there are too many searching nodes, which affect the computational efficiency and (2) the path obtained by the A* algorithm has redundant collinear nodes and redundant turning points. The robot walking along such a path will affect its movement continuity.

In view of the above problems, this paper proposes two improvements to the A* algorithm: (1) in order to improve the searching efficiency, an optimized search point selection strategy is proposed, and the heuristic function is improved; (2) in order to further optimize the path, the redundant point deletion strategy is introduced. Redundant nodes are eliminated so that the final path contains only the starting point, end point, and key turning points.

3.2.1. Optimization of Search Point Selection Strategy. The traditional A* algorithm expands the eight neighborhood grids of the current node during path planning. As shown in Figure 1, the pink grid is the current point, and n_1 to n_8 are the eight directions in which the current grid can move. Too many options of the orientation for the target point will cause unnecessary grids in the eight-neighbor grids, resulting in a waste of computing time and storage space.

In order to further improve the searching efficiency, an optimized search point selection strategy is introduced. According to the relative position of the target point and the current point, three search directions are discarded, and five search directions are retained. Let the angle between the segment from the target point to the current point and the direction of n_2 as ε , then the corresponding relationship between the angle ε and the 3 discarded directions is shown in Table 1. After the above optimization, the number of search nodes can be reduced, and the computational efficiency can be improved.

3.2.2. Heuristic Function of the Optimization Algorithm. The traditional A* algorithm will traverse many unnecessary nodes during the path searching process, which affects the searching efficiency. The key factor causing this problem is the heuristic function design of the A* algorithm. If the estimated value of the heuristic function is less than the actual cost value, too many search nodes will be generated, reducing the computational efficiency, though the optimal path can be obtained. However, if the estimated value of the heuristic function is greater than the actual cost value, a relatively

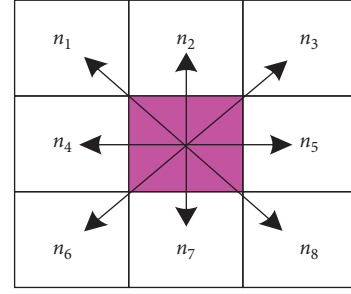


FIGURE 1: Schematic plot of the robot movement.

TABLE 1: The relationship between angle ε and the discarded directions.

ε	Retained directions	Discarded directions
$[337.5^\circ, 360^\circ) \cup [0^\circ, 22.5^\circ)$	$n_1n_2n_3n_4n_5$	$n_6n_7n_8$
$[22.5^\circ, 67.5^\circ)$	$n_1n_2n_3n_5n_8$	$n_4n_6n_7$
$[67.5^\circ, 112.5^\circ)$	$n_2n_3n_5n_7n_8$	$n_6n_7n_8$
$[112.5^\circ, 157.5^\circ)$	$n_3n_5n_6n_7n_8$	$n_1n_4n_6$
$[157.5^\circ, 202.5^\circ)$	$n_4n_5n_6n_7n_8$	$n_1n_2n_4$
$[202.5^\circ, 247.5^\circ)$	$n_1n_4n_6n_7n_8$	$n_2n_3n_5$
$[247.5^\circ, 292.5^\circ)$	$n_1n_2n_4n_6n_7$	$n_3n_5n_8$
$[292.5^\circ, 337.5^\circ)$	$n_1n_2n_3n_4n_6$	$n_5n_7n_8$

smaller number of searching nodes will be generated, which will improve efficiency, but it is difficult to obtain an optimal path. When the estimated value of the heuristic function is equal to the actual cost value, the highest search efficiency can be achieved. Since the heuristic function adopts Euclidean distance, the value of the heuristic function will always be less than or equal to the actual distance from the current point to the target point. When the current point is far from the target point, the estimated value of the heuristic function is much smaller than the actual value, which will cause the algorithm to search for more nodes and reduce efficiency. At this time, the weight of the estimated value should be increased to improve efficiency. When the current point is gradually approaching the target point, the estimated value gradually approaches the actual value. In order to prevent the estimated value from being too large and to avoid failure in searching for the optimal path, the estimated value weight should be reduced accordingly. In summary, we propose to improve the cost function as follows:

$$f(n) = g(n) + \left(1 + \frac{r}{R}\right)h(n), \quad (2)$$

where $f(n)$ is the integrated cost value, $h(n)$ is the estimated cost value from the current point to the target point, $g(n)$ is the actual cost value from the starting point to the current point, R is the distance from the starting point to the target point, and r is the distance between the current point and the target point.

3.2.3. Multiobjective Optimization. The traditional A* algorithm is only suitable for searching a single target point, and it has the shortcomings of low searching efficiency and

poor compatibility. Nowadays, the work situation is complicated, and the tracking of multiple target points in one planning can greatly improve efficiency. Therefore, the proposed method first samples and compares the location information of the target points and then improves the searching algorithm to consider multiple target points in one planning.

The priority of different target points is an important prerequisite for multitarget path planning. The Euclidean distance is used as the cost function of selecting the optimal target point. The lower the selection cost, the higher the priority level. Specific steps are as follows:

(1) Calculate the cost function of the target points T_i , $i = 1, 2, \dots, n$; (2) compare the priority levels of the target points T_i ; (3) preferentially plan the path for the target point T_i with the highest priority; (4) take the current target point T_{i-1} as the starting point of the mobile robot and plan the path of the secondary target point T_i ; and (5) judge whether to reach the n -th target point. If it arrives, stop searching; otherwise, go back to the last step.

4. Enhanced Dynamic Window Approach

In this paper, an enhanced dynamic window algorithm (EDWA) is proposed, which combines global path information to avoid obstacles and pursue dynamic targets. Even part of the environment information is unknown, the mobile robot can still perform local path planning, avoid obstacles, and reach the designated target point.

4.1. Kinematics Model of Mobile Robot. DWA mainly samples the speed space of the mobile robot in the window area and simulates the feasible motion trajectory within time t in the speed space (v_t, ω_t) . The changes of the linear velocity v_t and the angular velocity ω_t in the velocity space represent the motion states of the mobile robot. Among all feasible trajectories, the optimal trajectory is obtained with the evaluation function. Therefore, assuming that the mobile robot moves within the time interval Δ_t , the kinematics model can be formulated as follows [22]:

$$\begin{cases} x = x + v_x \Delta t \cos \theta_t - v_y \Delta t \sin \theta_t, \\ y = y + v_x \Delta t \sin \theta_t + v_y \Delta t \cos \theta_t, \\ \theta_t = \theta_t + \omega_t \Delta t. \end{cases} \quad (3)$$

4.2. Speed Sampling of Mobile Robot. There are an infinite number of groups in the velocity space, but in actual operation, it is necessary to constrain the sampling velocity range based on the constraints of the mobile robot as well as environmental constraints.

The speed constraints of mobile robots can be expressed as follows:

$$V_m = \{(v, \omega) | v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\}. \quad (4)$$

In the moving time interval of the dynamic window, the speed constraint of the mobile robot caused by the addition

and subtraction constraints of the motor can be calculated as follows:

$$V_d = \{(v, \omega) | v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t], \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t]\}, \quad (5)$$

where v_c and ω_c denote current speed, \dot{v}_a and $\dot{\omega}_a$ denote the maximum acceleration of the mobile robot, and \dot{v}_b and $\dot{\omega}_b$ denote the maximum deceleration of the mobile robot.

Mobile robot braking distance constraint: when avoiding obstacles in a local environment, the safety of the mobile robot needs to be ensured. Under the constraint of maximum deceleration, the speed can be reduced to 0 m/s before the impact. The braking constraint is expressed as follows:

$$V_d = \{(v, \omega) | v \leq (2d(v, \omega)\dot{v}_b)^{1/2}, \omega \leq (2d(v, \omega)\dot{\omega}_b)^{1/2}\}, \quad (6)$$

where $d(v, \omega)$ represents the shortest distance between the trajectory (v, ω) and the obstacle.

4.3. Evaluation Function Considering Global Path Information. After simulating several trajectories based on the range of motion speed and the robot motion model, the optimal trajectory is obtained through the evaluation function. The traditional DWA often falls into the local optimum. In order to fix this problem, an evaluation function considering the global path information is proposed to ensure that the final local path is in accordance with the global optimal path. The improved evaluation function is as follows:

$$G(v, \omega) = \sigma(\alpha PH(v, \omega) + \beta d(v, \omega) + \gamma v_e(v, \omega)), \quad (7)$$

where $PH(v, \omega)$ is the angle difference between the directions of the trajectory end points and the current target point, $d(v, \omega)$ is the distance between the trajectory and the nearest obstacle, $v_e(v, \omega)$ is the current speed evaluation function, σ is the smoothing function, and α , β , and λ are weighting coefficients. With the evaluation function, the path planned by the robot can avoid random obstacles and fit the global optimal path.

4.4. Dynamic Target Pursuing with Preview Deviation Angle Algorithm. In the current complex working environment, it is necessary for mobile robots to realize online path planning while avoiding obstacles and complete the task of pursuing dynamic targets as quickly as possible. In the process of pursuing the target, not only the dynamic path planning problem needs to be solved, but also the mobile robot needs to use its own sensors to continuously adjust the yaw angle and combine the weighted obstacle search algorithm to improve the chasing efficiency.

During the operation, the position information of the target point is constantly updated with the robot sensors; the center point of the target is regarded as the preview point; and the distance between the center of the mobile robot and the center of the target is the preview distance. The angle between the moving directions of the target point and the

mobile robot is the preview angle. The schematic diagram of chasing strategy based on the preview deviation angle is shown in Figure 2. In Figure 2, L is the preview distance, and M_d is the vertical distance between the mobile robot and the dynamic target point trajectory. Given $L \neq 0$ and $M_d \neq 0$, then M_d is always less than L , θ_1 is the preview deviation angle, and it is stipulated that the preview deviation angle is positive when the preview point is in the front-left of the moving direction of the mobile robot, and the center of the moving target point is the preview point.

It can be seen from Figure 2 that when $\theta_1 \rightarrow 0$ and $L \rightarrow 0$, the mobile robot can catch up with the moving target, so we get:

$$\phi_1 = \arctan \frac{M_d}{N} - \theta_1. \quad (8)$$

In the case of $L \neq 0$ and $M_d = 0$, it can be seen from Figure 2 that the robot is always behind the target point during the pursuit; we get $N = L$, $\arctan(M_d/N) = 0$, then $\theta_1 = 0$, and $\phi_1 = 0$; and the robot accelerates in a straight line to chase the target point. In the case of $L \neq 0$ and $M_d \neq 0$, it can be derived that during the pursuing process $N \neq 0$, $\arctan(M_d/N) \rightarrow 0$ and $\theta_1 \rightarrow 0$, so $\phi_1 \rightarrow 0$. In Figure 2, if an arc is drawn with O as the center and tangent to the trajectory of the moving target, then the arc is the planned trajectory, and the arc radius R can be calculated as follows:

$$R = \frac{L^2}{2M_d}. \quad (9)$$

From the sensor readings of the mobile robot, the coordinates of the moving target point (x_g, y_g) and the mobile robot (x, y) can be measured, then the coordinates of the center $O(x_r, y_r)$ can be obtained. The center O that is closer to the mobile robot will be selected:

$$\begin{cases} -x + \tan \theta_g \cdot y - b \cdot \tan \theta_g = 0, \\ \left| \frac{1 - x_r + \tan \theta_g \cdot y_r - b \tan \theta_g}{\sqrt{1 + \tan^2 \theta_g}} \right| = R. \end{cases} \quad (10)$$

Then the arc trajectory is expressed as follows:

$$(x - x_r)^2 + (y - y_r)^2 = R^2. \quad (11)$$

5. Experiment

In order to verify the global and local path planning performance of the proposed method, we compare the proposed hybrid method with other methods in different environments. The experiment runs on a computer with a CPU of I5-9400 and a RAM of 16 GB, and the algorithm is implemented through MATLAB programming simulation. The parameters are set as follows: the maximum velocity is 1 m/s, the maximum angular velocity is 20°/s, the velocity resolution is 0.01 m/s, the angular velocity resolution is 1°/s, the acceleration is 0.2 m/s², and the angular acceleration is 50°/s². The parameters of the evaluation function are: $\alpha = 0.1$, $\beta = 0.05$, and $\gamma = 0.2$.

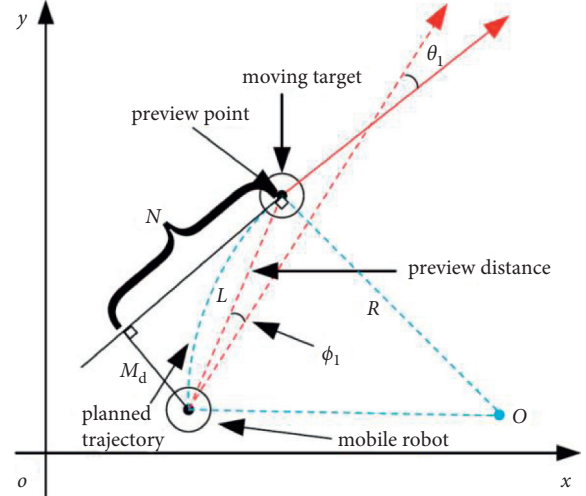


FIGURE 2: Preview deviation angle tracking.

5.1. Map and Environment Construction. In order to facilitate performance analysis, this article uses the following settings for mobile robots and environmental maps:

- (1) The grid method is used to mathematically model the environment map, and the feasible and infeasible areas are represented by white grids and black grids, respectively.
- (2) The starting point and ending point of the map are given, with some unknown obstacles in the map.
- (3) The mobile robot and dynamic obstacles are represented as dilated points.
- (4) The dynamic obstacle in the map moves in a straight line at a uniform speed, but the moving direction and position of the obstacle are unknown.
- (5) The mobile robot is equipped with sensors, which can perceive the information of the map within a limited range, such as the position and speed of obstacles.
- (6) The mobile robot moves at a constant speed and can move in all directions.

5.2. Trajectory Comparison. On the static global path planning problem, compared with the traditional A* algorithm, the proposed algorithm mainly optimizes the planned path trajectory. In a 20×20 grid environment, global path planning is performed with (1, 1) as the starting point and (20, 20) as the ending point, and the performance of the proposed method is compared with the classic A* algorithm [18]. The reference standards for optimizing the trajectory include the cumulative turning angles, the average turning angles, and the path length. Figure 3 shows the optimization process of path nodes, and the red line represents the generated global path. Figure 3(a) is the path trajectory generated by the traditional A* algorithm, Figure 3(b) is the path trajectory generated with [18], and Figure 3(c) is the path trajectory generated by the proposed method. The statistics are shown in Table 2.

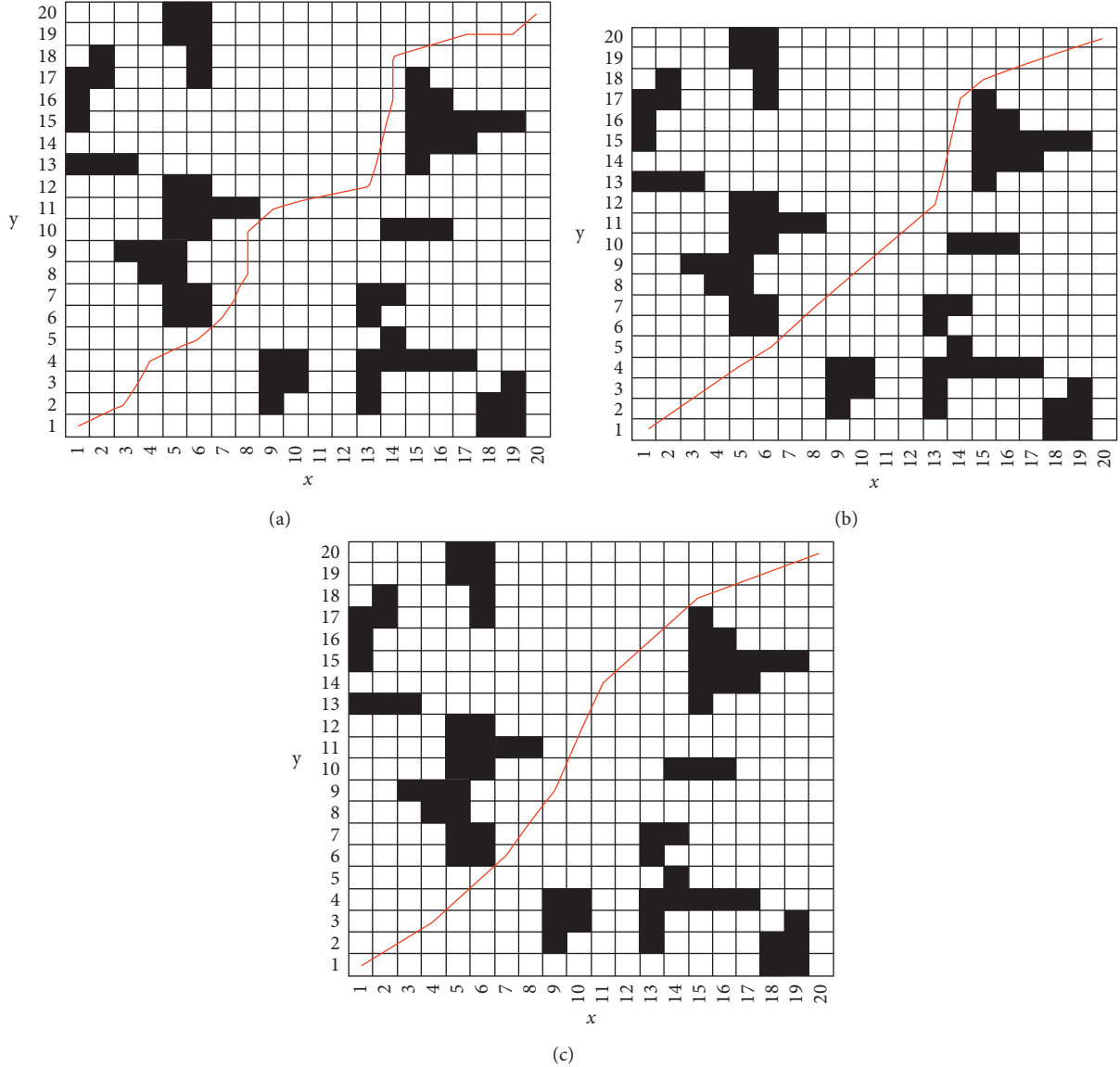


FIGURE 3: Path planning results. (a) The path trajectory generated by the traditional A* algorithm. (b) The path trajectory generated with [18]. (c) The path trajectory generated by the proposed method.

TABLE 2: Planned paths with different methods.

Method	Nodes	Cumulative turning angles (°)	Average turning angles (°)	Path length
Classic A*	15	488.49	32.56	33.77
[18]	5	128.92	25.78	25.89
Proposed method	6	85.41	14.24	26.05

It can be seen from Table 2 that compared with the traditional A* algorithm, the proposed method has significantly reduced the cumulative turning angles, average turning angles, and path length through path optimization strategy. The method in [18] only deletes redundant points for the trajectory. It can be seen from Table 2 that although the proposed algorithm has a slight increase in path length compared to [18], the cumulative turning angle and average turning angle have been reduced, making the planned trajectory smoother and more in line with the kinematics law of

the mobile robot. In addition, the multiobjective strategy adopted in this paper shortens the average distance between nodes, which is more conducive to the selection of the subsequent local path planning range.

5.3. Comparison of Algorithm Efficiency. In order to alleviate the impact of environmental changes on the experiments and further improve the evaluation fairness, different data are used to simulate different algorithms on the same

TABLE 3: Comparison under different environments.

Method	20 × 20		50 × 50		100 × 100	
	Path planning time (T)	Path length	Path planning time (T)	Path length	Path planning time (T)	Path length
Classic A* [18]	22	28.7	26	82.8	155	169.7
Proposed method	10	26.1	16	61.9	42	74.8

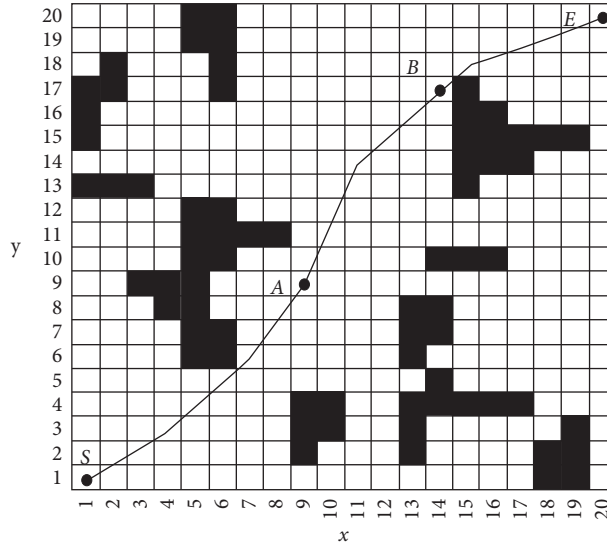


FIGURE 4: Path planned under given environment.

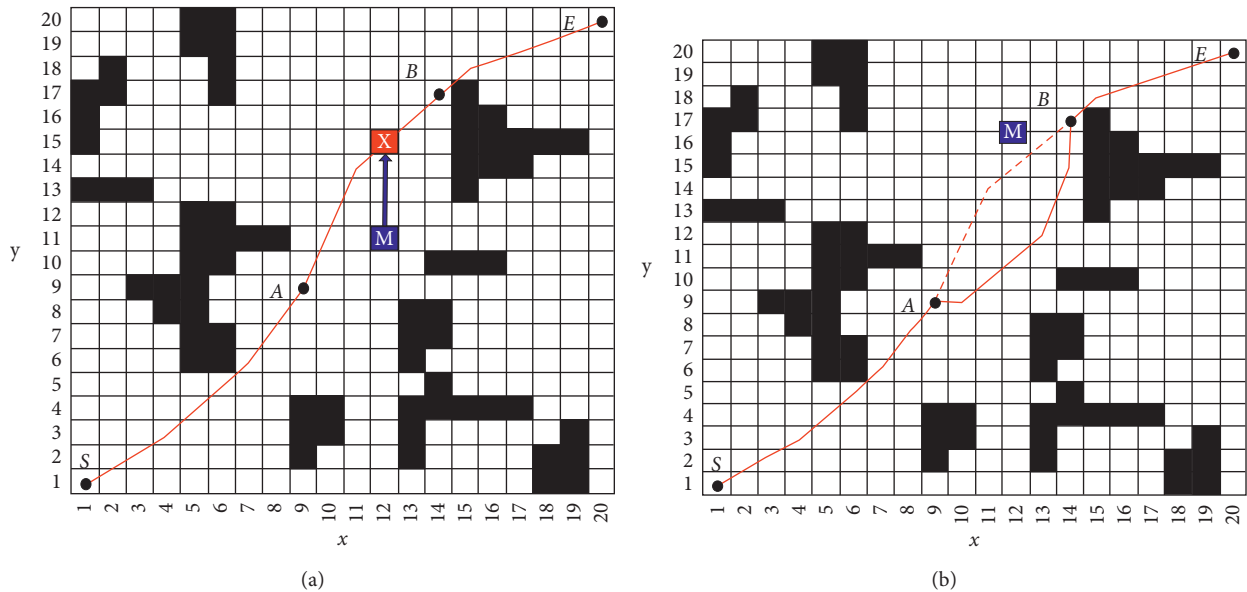


FIGURE 5: Dynamic obstacle avoidance: (a) the robot is located in A and (b) the robot is located in B.

computer, and three environments of different complexity are established. The complexity of the environment is set as the size of the environment and the coverage of obstacles.

Table 3 compares the proposed algorithm with other methods from the path planning time (the number of cycles T) and the planned path length. From the results, it can be

found that the proposed method has achieved good results in various environments and significantly reduced the computational complexity. The reason is that in the proposed method, a lot of the search nodes have been simplified compared with the traditional A* algorithm and the method in [18]. In terms of the number of turning points, the

TABLE 4: Results under dynamic environment.

Method	Reach destination	Planning time (T)	Path length
Classic A* [18]	No	—	—
Proposed method	Yes	35	55.8
Proposed method	Yes	26	47.6

proposed algorithm can reduce the number of turning points in most cases. And through the improvement of multiobjective optimization and heuristic function, the time and space cost of path planning is greatly reduced.

5.4. Dynamic Environment Simulation. In order to verify the dynamic programming ability of the proposed method, unknown dynamic obstacles are introduced in the simulation, and a simple environment is established for simulation verification. In this environment, according to the given map information, with $S(1, 1)$ as the starting point and $E(20, 20)$ as the end point, global path planning is performed. After optimization, the path shown in Figure 4 is obtained, which is represented by a red line. Select appropriate local subtarget points on the path and add two local subtarget points A and B.

In this environment, the path is divided into three subpaths, namely SA, AB, and BE. When the robot is at the starting position, the DWA does not detect any changes in the map, and the robot travels from point S to the first local subtarget point A along the path. At this time, the robot detects a dynamic obstacle M (blue square). According to the information feedback from the sensors, the robot will collide with the obstacle at the X position (red grid), as shown in Figure 5(a).

According to the sensor information, the traveling speed of the dynamic obstacle M is less than the robot, and the moving trajectory of M has an intersection with the path trajectory of the robot. The EDWA is used to replan the path in the local area and update the subpath AB. As shown in Figure 5(b), the original path is represented by a dashed line, and the path planned by the proposed method is represented by a solid line. When the second subtarget point B is reached, the obstacle is located on the left side of the robot, completing the dynamic obstacle avoidance process.

Table 4 shows the quantitative results of different algorithms when unknown dynamic obstacles are introduced in the 20×20 grid environment shown in Figure 4. The traditional A* algorithm does not have the ability to avoid obstacles. After encountering unknown obstacles, the path planning fails, and the target location cannot be reached. The proposed algorithm and [18] both have the ability to replan the path when encountering unknown obstacles, and both reach the target position. The processing time and path length of the proposed algorithm are better than those of [18].

Figure 6 shows the simulation results of the proposed method for avoiding dynamic obstacles and chasing dynamic target points. The proposed hybrid algorithm can realize local path planning and at the same time use the preview deviation angle algorithm to complete the pursuit of dynamic target points. And after combining the global path information, the hybrid algorithm plans a path with a shorter length and higher smoothness.

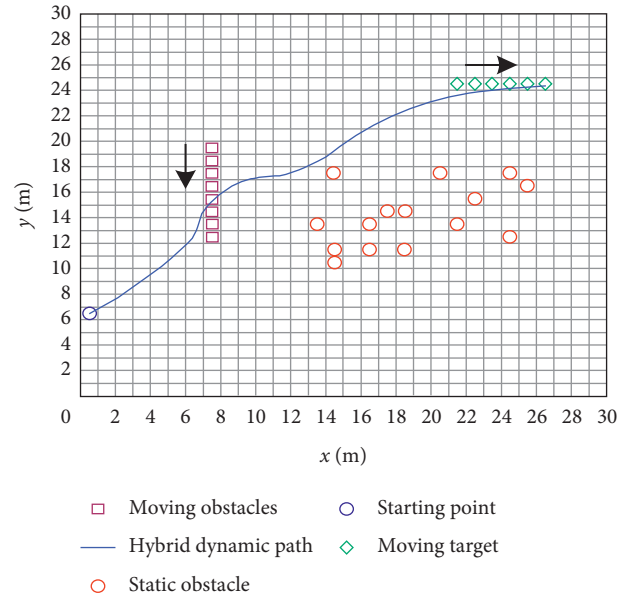


FIGURE 6: Path planning with dynamic obstacles and moving target.

6. Conclusion

This paper proposes a random obstacle avoidance method for robot path planning that combines the improved A* algorithm with EDWA. By optimizing the search point selection strategy and evaluation function of the traditional A* algorithm, the search efficiency is improved, and a redundant node deletion strategy is introduced to reduce the path length. The EDWA is used to realize the real-time path planning and make up for the shortcomings of the traditional A* algorithm with poor timeliness. While chasing dynamic obstacles, the optimal path is planned in accordance with the global path planning information. Through comparative analysis, the effectiveness of the proposed hybrid algorithm is proved. In the future, we plan to incorporate machine vision techniques to further explore the path planning of mobile robots in complex environments and multitask conditions.

Data Availability

The data included in this paper are available without any restriction.

Conflicts of Interest

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

References

- [1] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: on path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.
- [2] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: concepts, methods, theoretical framework, and

- applications,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, Article ID 1729881419839596, 2019.
- [3] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, “Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics,” *Annals of Operations Research*, vol. 308, pp. 1–19, 2020.
- [4] A. Pandey, A. K. Kashyap, D. R. Parhi, and B. K. Patle, “Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture,” *World Journal of Engineering*, vol. 16, 2019.
- [5] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, “Substantial capabilities of robotics in enhancing industry 4.0 implementation,” *Cognitive Robotics*, vol. 1, 2021.
- [6] H. Jahanshahi and N. N. Sari, “Robot path planning algorithms: a review of theory and experiment,” 2018, <https://arxiv.org/abs/1805.08137>.
- [7] B. K. Patle, D. R. K. Parhi, A. Jagadeesh, and S. K. Kashyap, “Matrix-binary codes based genetic algorithm for path planning of mobile robot,” *Computers & Electrical Engineering*, vol. 67, pp. 708–728, 2018.
- [8] E. S. Low, P. Ong, and K. C. Cheah, “Solving the optimal path planning of a mobile robot using improved Q-learning,” *Robotics and Autonomous Systems*, vol. 115, pp. 143–161, 2019.
- [9] B. Li, H. Liu, and W. Su, “Topology optimization techniques for mobile robot path planning,” *Applied Soft Computing*, vol. 78, pp. 528–544, 2019.
- [10] M. Kusuma and C. Machbub, “Humanoid robot path planning and rerouting using A-star search algorithm,” in *Proceedings of the 2019 IEEE International Conference on Signals and Systems (ICSigSys)*, pp. 110–115, IEEE, Bandung, Indonesia, July 2019.
- [11] J.-H. Jung and D.-H. Kim, “Local path planning of a mobile robot using a novel grid-based potential method,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 20, no. 1, pp. 26–34, 2020.
- [12] A. Özdemir and V. Sezer, “A hybrid obstacle avoidance method: follow the gap with dynamic window approach,” in *Proceedings of the 2017 First IEEE International Conference on Robotic Computing (IRC)*, pp. 257–262, IEEE, Taichung, Taiwan, April 2017.
- [13] Q. Yuan and C.-S. Han, “Research on robot path planning based on smooth A* algorithm for different grid scale obstacle environment,” *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 8, pp. 5312–5321, 2016.
- [14] L. Yu, D. Kong, X. Shao, and X. Yan, “A path planning and navigation control system design for driverless electric bus,” *IEEE Access*, vol. 6, pp. 53960–53975, 2018.
- [15] A. Kaplan, N. Kingry, P. Uhing, and R. Dai, “Time-optimal path planning with power schedules for a solar-powered ground robot,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1235–1244, 2016.
- [16] D. Kiss and G. Tevesz, “Advanced dynamic window based navigation approach using model predictive control,” in *Proceedings of the 2012 17th International Conference on Methods & Models in Automation & Robotics (MMAR)*, pp. 148–153, IEEE, Miedzydroje, Poland, August 2012.
- [17] C. Henkel, A. Bubeck, and W. Xu, “Energy efficient dynamic window approach for local path planning in mobile service robotics,” *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 32–37, 2016.
- [18] X. Zhong, J. Tian, H. Hu, and X. Peng, “Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment,” *Journal of Intelligent and Robotic Systems*, vol. 99, no. 1, pp. 65–77, 2020.
- [19] Q. Wu, Z. Chen, L. Wang et al., “Real-time dynamic path planning of mobile robots: a novel hybrid heuristic optimization algorithm,” *Sensors*, vol. 20, no. 1, p. 188, 2020.
- [20] A. K. Kashyap, D. R. Parhi, M. K. Muni, and K. K. Pandey, “A hybrid technique for path planning of humanoid robot NAO in static and dynamic terrains,” *Applied Soft Computing*, vol. 96, Article ID 106581, 2020.
- [21] Z. Ma, H. Qiu, H. Wang, L. Yang, L. Huang, and R. Qiu, “A* algorithm path planning and minimum snap trajectory generation for mobile robot,” in *Proceedings of the 2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE)*, pp. 284–288, IEEE, Wuhan, China, December 2021.
- [22] L. Chang, L. Shan, C. Jiang, and Y. Dai, “Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment,” *Autonomous Robots*, vol. 45, no. 1, pp. 51–76, 2021.