

Optimizing obstacle avoidance path planning for intelligent mobile robots in multi-obstacle environments

Zhang, H.Q.^{a,*}

^aSchool of Electrical and Information Engineering, Jiangsu University of Technology, Changzhou, P.R. China

ABSTRACT

To enhance the obstacle avoidance capabilities of intelligent mobile robots in environments with multiple obstacles, a method for optimal path planning has been developed and analysed. Multi-obstacle scene maps were generated using the safety boundary condition method and envelope method, with obstacle distribution abstractly represented by an obstacle rate. The robot's pose and orientation were determined through the application of Adaptive Monte Carlo localization. Speed sampling was conducted to identify the optimal speed combination, establishing a set of candidate trajectories. A trajectory evaluation function was constructed, and the trajectory with the highest score was selected. The improved A* algorithm was integrated into the evaluation function, incorporating the obstacle rate to refine path planning. To eliminate redundant sections, a key point extraction algorithm was employed to adjust and smooth the path, ensuring optimal obstacle avoidance. The proposed approach demonstrated superior performance, achieving a shorter planned path, a rapid obstacle avoidance reaction time of 32 milliseconds, a 100 % success rate in obstacle avoidance, and a path smoothness of 97 %. These results indicate that the method is highly effective for obstacle avoidance, with significant improvements in both safety and efficiency.

ARTICLE INFO

Keywords:

Intelligent mobile robots;
Obstacle avoidance path planning;
Adaptive Monte Carlo;
Envelope method

*Corresponding author:

zhq@jsut.edu.cn
(Zhang, H.Q.)

Article history:

Received 5 September 2024

Revised 20 October 2024

Accepted 27 October 2024



Content from this work may be used under the terms of the Creative Commons Attribution 4.0 International Licence (CC BY 4.0). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

1. Introduction

With the continuous progress and innovation of science and technology, mobile robot technology has become one of the hotspots in today's research field [1-8], which has a wide range of applications, including but not limited to industrial manufacturing, logistics distribution, medical care, rescue exploration, etc. [9, 10]. In these complex and dynamic environments, mobile robots need to be able to quickly and accurately plan the optimal or feasible path from the starting point to the target point while avoiding various obstacles. As the core issue of mobile robot research, path planning has far-reaching significance and wide application value. It is not only about how the robot reaches the designated destination from a starting point but also an important embodiment of the robot's autonomous navigation, environmental perception, and decision-making ability. In the process of solving this problem, it is necessary to fully consider the complex factors, such as the motion characteristics of the robot, the determination of the starting and target positions, and the possible obstacles in the surrounding environment, so as to plan a safe, effective, and optimized path. With continuous progress in the field of artificial intelligence, robots have become increasingly capable of environmental perception and decision-making [3]. However, how to achieve optimal obstacle avoidance path planning in complex and changeable environments is still a major challenge in the field of robotics. Carrying out relevant

research not only helps to promote the progress of related technologies but also provides more possibilities for the practical application of robot technology.

Therefore, relevant research scholars have carried out research on this. For example, Li *et al.* [10] proposed an improved nature-inspired swarm intelligence (NISI) meta-heuristic algorithm called the Slime Mold Optimization Algorithm (SMOA), which was inspired by the oscillation characteristics of slime moulds when they met prey. The ability to read workspace maps and plan collision-free paths with minimal processing time and cost under constraints. However, the SMOA algorithm may be optimized for specific types of environments or obstacle layouts at the beginning of design. In practical applications, robots may face a variety of complex and dynamic environments, including different types of obstacles, irregular shapes and sizes, and dynamic obstacles. If the SMOA algorithm cannot adapt to these changes well, its performance may be affected, which makes it difficult for the robot to effectively find the optimal obstacle avoidance path. Chen *et al.* [11] proposed a three-neighbourhood search A* algorithm combined with an artificial potential field to optimize the path planning of mobile robots. The algorithm integrates and improves part of the artificial potential field and the A* algorithm to solve the irregular obstacles in the forward direction. In the process of obstacle avoidance, the current pose vector of the mobile robot is constructed, which reduces the search scope to less than three neighbours and avoids repeated searches. However, in the practical application of this method, if there are narrow channels or dense areas of obstacles in the environment, the three-neighbourhood search strategy may make the robot fall into local optimization, and the effect is not good. Mohanraj *et al.* [12] developed a hybrid algorithm to find the approximate optimal solution of global path planning and enhanced the algorithm by considering the improved region and real-time constraints to avoid collision with obstacles and find the best path. However, this algorithm is computationally burdensome, especially in large-scale or complex environments, which leads to long computing times and cannot achieve the desired real-time performance. Chen *et al.* [13] proposed an artificial potential field method based on global path guidance to solve the problem of target inaccessibility and local minima of the traditional artificial potential field method in complex environments with dynamic obstacles. Firstly, to solve the problem of target inaccessibility, global path attraction is added to APF. Secondly, an obstacle detection optimization method is proposed, which selects the optimal virtual target point by setting the evaluation function to improve the local minimum problem. Finally, based on the optimization method of obstacle detection, the processes of gravity and repulsion are improved so that the path can pass through the narrow channel smoothly and keep collision-free. Because different environments and tasks may require different evaluation functions and parameter settings, if there are dynamic obstacles in the environment or the shape and size of obstacles are complex and changeable, the performance of obstacle detection and optimization methods may be affected, resulting in the robot being unable to accurately identify and avoid obstacles.

In order to further improve the effect of path planning of intelligent mobile robots, the optimal obstacle avoidance path planning of intelligent mobile robots in multi-obstacle scenes is proposed. The innovative technological roadmap of this method is as follows:

- Environmental modelling: Using the method of adding safety boundary conditions and the envelope method to construct a multi obstacle scene map, setting safety boundaries for each obstacle, and enveloping complex shaped obstacles with simple geometric shapes. Simultaneously calculate the obstacle rate, which is obtained by the ratio of obstacle area to the total map area, to abstractly represent the distribution of obstacles.
- Robot localization: The adaptive Monte Carlo localization (AMCL) algorithm is used to determine the pose and direction of the robot. Using scattered particles to represent the possible position and direction of the robot, updating particle weights based on sensor information, and adaptively adjusting the number and distribution of particles based on the robot's motion state and sensor reliability.
- Trajectory generation speed sampling and candidate trajectories: Based on the current pose and positioning information of the robot, speed sampling is performed, and speed values are selected according to rules within the feasible speed range. Based on the robot

motion model, predict the motion trajectory for each sampling speed within a certain period of time and construct a candidate trajectory set.

- Trajectory generation evaluation and selection: Construct a trajectory evaluation function that comprehensively considers factors such as the distance between the trajectory and obstacles (reference obstacle rate), trajectory length, and motion smoothness. Apply this function to the candidate trajectory set and select the trajectory with the highest score as the preliminary optimal trajectory.
- Path optimization: Improve the A* algorithm by introducing obstacle rates into the evaluation function, making the algorithm more focused on avoiding areas with dense obstacles. Integrate key point extraction algorithms to identify and eliminate redundant road segments. Finally, the optimized path is recombined with the preliminary optimal trajectory to obtain the final optimal obstacle avoidance path planning result.

2. Multi-obstacle scene map generation

In an environment of irregular obstacles, irregular obstacles are assumed to be two types of obstacles, namely convex polygon obstacles and concave polygon obstacles. There are two methods to deal with these two kinds of obstacles, which can be transformed into regular obstacles by adding the safety boundary condition method and the envelope method [14].

For complex scenes, in addition to environmental factors, the irregularity of obstacles is the main influencing factor. To optimize the path, especially to judge whether the robot can move in an irregular, complex obstacle environment, it is necessary to obtain the size of the safety boundary conditions and the robot body, and build the safety distance model d_{safe} as follows:

$$d_{safe} = r + l \quad (1)$$

where, d_{safe} is the safe distance, r is the working radius of the robot l , is an important constant. The value of l should neither be too small (which may cause collisions with obstacles) nor too large (which may hinder the ability to find an optimal path). After adding safety boundary conditions to irregular multi-obstacles, the robot can be imagined as a particle, which can effectively prevent the robot from touching obstacles, as shown in Fig. 1.

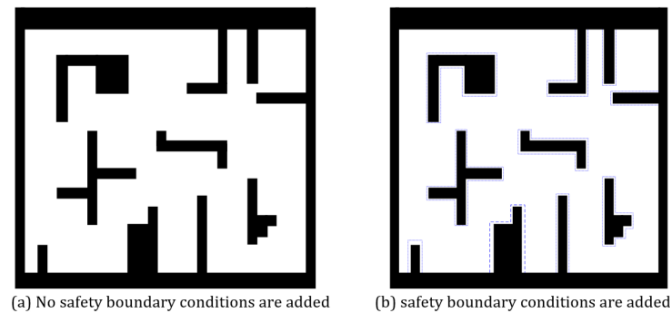


Fig. 1 Multi obstacle scene map

As shown in Fig. 1, in generating a multi-obstacle scene map, it is necessary to quantify its relevant information. The traditional A* algorithm for path planning tends to increase the search space and reduce efficiency, especially in environments with a large number of obstacles [15, 16]. If the obstacle information can be integrated into the evaluation function and the heuristic function can be dynamically adjusted according to the distribution of obstacles, the search space can be greatly reduced, and the search efficiency can be improved. Because the distribution of obstacles on different maps is also different, it is necessary to formulate a standard to represent the distribution of obstacles.

Set an obstacle rate P in the grid map to describe the distribution of obstacles. Set the starting node coordinate as (x_s, y_s) , the target node coordinate as (x_g, y_g) , and the number of obstacle grids as N , then the expression of obstacle rate is as follows:

$$P = \frac{N}{(|x_s - x_g| + 1) \times (|y_s - y_g| + 1)} (P \in (0,1)) \quad (2)$$

3. Positioning and speed sampling

When the robot is in motion, the speed is related to the problem of minimizing time consumption, and the angle is related to the obstacle. Generally speaking, both are also closely related to the path optimization problem. In this paper, the fuzzy logic strategy is introduced. By adjusting the speed and angle of the robot, an asymptotically optimal and smooth path is planned, which makes the path more in line with the dynamic stability of the robot.

3.1 Adaptive Monte Carlo localization

The basic idea of robot obstacle avoidance is to ensure that it is close to the target while avoiding obstacles. The fuzzy logic strategy is realized by the fuzzy logic controller. The fuzzy logic controller is activated when the robot's obstacle sensor detects any obstacle. The fuzzy logic controller will produce corresponding changes in the speed and angle of the robot according to different conditions of the environment [17, 18]. The robot abduction problem refers to the situation where, even though the robot's position and posture are initially known, they are altered due to external interference. As a result, the odometer may provide inaccurate data or fail to offer any control information altogether. The adaptive Monte Carlo localization method solves the problem that the robot cannot use the odometer model to update the particle swarm due to the change of position and posture caused by external forces during the movement of the robot, which eventually leads to the failure of localization [19, 20]. To solve this problem, the adaptive Monte Carlo positioning method randomly scatters particles with a certain probability in the grid map, and the calculation formula is as follows:

$$\begin{cases} w_{slow} = w_s + \partial_{slow}(w_{avg} - w_{slow}) \\ w_{fast} = w_f + \partial_{fast}(w_{avg} - w_{fast}) \end{cases} \quad (3)$$

where, w_{slow} is the long-term estimated weight, w_{fast} is the short-term estimated weight, w_{avg} is the average weight value of the particle swarm, and ∂_{fast} and ∂_{slow} generally take the default values of 0.1 and 0.001.

The formula for calculating the probability of randomly sowing seeds is as follows:

$$\max\left\{0, 1 - \frac{w_{fast}}{w_{slow}}\right\} \quad (4)$$

In the kidnapping problem of mobile robot, the weight value at this time is calculated with the observation value of lidar, the value of w_{avg} will be greatly reduced, and w_{fast} will also be far less than the value of w_{slow} . At this time, the particle swarm will appear randomly in the map, and then focus on global positioning. Finally, the particle swarm will match the real position.

Another advantage of adaptive Monte Carlo localization is that it uses the KLD sampling method, which can adjust the number of particles generated autonomously, so that the efficiency of particle filtering can be further improved. The calculation formula of the maximum number of particles M_x is as follows:

$$M_x = \frac{k-1}{2\varepsilon} \left(1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} Z_{1-\delta} \right)^3 \quad (5)$$

where, k is the number of nonempty particle state spaces, $Z_{1-\delta}$ is the $1 - \delta$ quantile on the normal distribution, and ε is the given value. During initialization, because particles are scattered due to random seeding, the value of k is relatively large, and the required value of M_x is also large. After a long time of continuous updating and convergence, the value of k will gradually decrease, and the value of M_x will also decrease. This summarizes the way of automatically adjusting the number of particles, reduces the amount of calculation, improves efficiency, and shortens the time of the algorithm.

Based on the above, the robot's position and pose positioning in the grid map is completed. Then, to plan an asymptotically optimal and smooth path to make the path more consistent with the stability of the robot's dynamics, the speed and angle of the robot's movement are obtained to determine its direction. Therefore, according to the kinematics formula, the kinematics model of the robot is constructed, which can be expressed as:

$$\dot{v} = \frac{1}{\tau_v} V + \frac{1}{\tau_v} v_c \quad (6)$$

$$\ddot{\theta} = \frac{1}{\tau_\theta} \dot{\theta} + \frac{1}{\tau_\theta} \theta_c \quad (7)$$

where, v_c is the motion speed of the robot, θ_c is the direction angle of the robot, τ_v is the time speed constant of the robot's autonomous motion, V is the initial speed of the robot, τ_θ is the time constant of the angular speed, and θ is the initial angular speed of the robot.

Set the distance ρ from the mobile robot to the starting position. At this time, the relative distance from the mobile robot to the starting position is:

$$x_0 = \rho \cos \theta \quad (8)$$

$$y_0 = \rho \sin \theta \quad (9)$$

where, x_0 is the horizontal distance from the mobile robot to the starting position, y_0 is the vertical distance from the mobile robot to the starting position, and the derivative of displacement $\rho' = v$ is obtained:

$$\dot{x}_0 = v \cos \theta - \rho \dot{\theta} \sin \theta \quad (10)$$

$$\dot{y}_0 = v \sin \theta - \rho \dot{\theta} \cos \theta \quad (11)$$

where, $\dot{x}_0$ is the horizontal speed of the mobile robot to the starting position, and $\dot{y}_0$ is the vertical speed of the mobile robot to the starting position.

To sum up, adaptive Monte Carlo is used to complete the positioning of the robot's pose and direction, so as to determine its position in the environment more accurately, and then find the optimal obstacle avoidance path from the starting point to the end point according to the environmental map and obstacle information.

3.2 Speed sampling

The precise control of speed allows the robot to effectively avoid obstacles, thereby enhancing the success rate of obstacle avoidance [21]. Thus, velocity sampling is performed to obtain the optimal velocity combination, and a candidate trajectory set is established based on the trajectory under this velocity constraint. Through subsequent evaluation, the target position can be achieved more efficiently.

In the two-dimensional velocity space, let the velocity set be (v_i, w_i) , where v_i is the linear velocity and w_i is the angular velocity. Because the mobile robot is constrained by its own indoor structure and the surrounding environment, there is a limit to the sampling speed, which mainly has the following three constraints:

- Maximum and minimum speed constraints. In practical applications, due to the limited hardware performance of mobile robots, there is a limit value to the speed. V_m is defined as the set of the maximum and minimum speed constraints of mobile robots, and V_m is expressed as:

$$V_m = \{(v, w) | v \in [v_{min}, v_{max}] \cap w \in [w_{MIN}, w_{max}]\} \quad (12)$$

where, v_{min} is the minimum linear velocity, v_{max} is the maximum linear velocity, w_{min} is the minimum angular velocity, and w_{max} is the maximum angular velocity.

- Motor acceleration and deceleration constraints. Due to the limited torque of the motor and other tools, there are restrictions on the acceleration and deceleration of the motor. V_d is defined as the set of the maximum and minimum speeds that the mobile robot can produce within the predicted time interval, and V_d is expressed as:

$$V_d = \{(v, w) | v \in [v_t - \dot{v}\Delta t, v_t + \dot{v}\Delta t] \cap w \in [w_t - \dot{w}\Delta t, w_t + \dot{w}\Delta t]\} \quad (13)$$

where, v_t is the linear velocity of the mobile robot at time t , w_t is the angular velocity at time t , $\dot{v}$ is the maximum linear velocity of the mobile robot, $\dot{w}$ is the maximum angular velocity, and Δt is the simulation prediction time interval.

- Braking distance constraint. This constraint requires the robot to effectively reduce its speed to zero before the actual collision according to its maximum deceleration when facing the potential collision risk, to avoid any form of collision accident. Define V_a as the speed set of the braking distance constraint, then V_a is represented as:

$$V_a = \{(v, w) \mid v \leq \sqrt{2 \text{dist}(v, w) \cdot \dot{v}} \cap w \leq \sqrt{2 \text{dist}(v, w) \cdot \dot{w}}\} \quad (14)$$

where, $\text{dist}(v, w)$ is the nearest distance from the mobile robot to the obstacle in state (v, w) . Therefore, the speed sampling interval V of the mobile robot in time period Δt is:

$$V = V_m \cap V_d \cap V_a \quad (15)$$

After obtaining the velocity sampling spatial model, according to the current state and environmental information of the mobile robot, the possible trajectories under constraint conditions are predicted to establish a candidate trajectory set. Calculate the cost of the trajectory to score, and select the trajectory with the highest score as the next movement scheme of the mobile robot.

4. Optimal obstacle avoidance path planning for mobile robot

4.1 Trajectory evaluation function

After determining the motion model of the mobile robot and the velocity sampling space V , several feasible trajectories are simulated through multiple groups of selectable velocity combinations in the velocity sampling space. At this time, it is necessary to establish a trajectory evaluation function for selecting the optimal trajectory from these feasible trajectories. In the dynamic serial port algorithm, the trajectory evaluation function consists of the target azimuth evaluation function, the obstacle clearance evaluation function, and the speed evaluation function, as follows:

$$G(v, \omega) = \varepsilon(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dis}(v, \omega) + \gamma \cdot \text{vel}(v, \omega)) \quad (16)$$

where, $\text{heading}(v, \omega)$ is the target azimuth evaluation function, $\text{dis}(v, \omega)$ is the obstacle clearance evaluation function, $\text{vel}(v, \omega)$ is the velocity evaluation function, α , β and γ are the weights of the target azimuth evaluation function, obstacle clearance evaluation function, and velocity evaluation function respectively, and ε is the normalization coefficient.

- The target azimuth evaluation function $\text{heading}(v, \omega)$ is used to measure the degree to which the end point of the simulated trajectory of the mobile robot faces the target point, as shown in Eq. 17. The included angle between the direction of the end point of the mobile robot trajectory and the connecting line of the target point is θ . The smaller the θ value, the more the mobile robot tends to move towards the target point, and the larger $\text{heading}(v, \omega)$.

$$\text{heading}(v, \omega) = 180^\circ - \theta \quad (17)$$

- The obstacle clearance function $\text{dis}(v, \omega)$ represents the distance from the point on the simulated trajectory of the mobile robot to the nearest obstacle. The larger the value, the farther the mobile robot is from the obstacle on the simulated trajectory, and the higher the safety of mobile robot navigation [22]. The function of $d_{\max}$ in Eq. 18 is to avoid the absence of obstacles on a simulated trajectory, which leads to an excessive gap function of obstacles and accounts for too much in the total trajectory evaluation function.

$$\text{dis}(v, \omega) = \begin{cases} d, & d < d_{\max} \\ d_{\max}, & d \geq d_{\max} \end{cases} \quad (18)$$

where, d is the distance from the point on the trajectory of the mobile robot to the nearest obstacle, and $d_{\max}$ is the threshold value of the distance from the mobile robot to the obstacle.

- The speed evaluation function $vel(v, \omega)$ is used to evaluate the forward speed of the robot in the process of moving towards the target point, as shown in Eq. 19. The larger the value of $vel(v, \omega)$, the faster the mobile robot can approach the target point on the simulated trajectory.

$$vel(v, \omega) = |v_g| \quad (19)$$

where, v_g is the linear velocity corresponding to the simulated trajectory of the mobile robot.

4.2 Improved evaluation function

The heuristic function of A* algorithm can be dynamically adjusted according to the distribution of obstacles. When there are fewer obstacles, the heuristic function will be increased, the search space will be reduced, and the operation speed will be improved [23]. When there are many obstacles, the heuristic function cannot be increased all the time, otherwise, the algorithm will fall into local optimization, and there will be unnecessary turns, which will increase the path cost and make it difficult to search for the optimal path. Therefore, when there are many obstacles, the value of the heuristic function is reduced to make the path more accurate. The search space of the A* algorithm changes with the position of the starting node and the target node [24, 25]. The improved A* algorithm introduces the obstacle rate P into the evaluation function, as shown in Eq. 20, and changes the weights of $g(n)$ and $h(n)$ according to the obstacle rate P , so as to improve the adaptability of the evaluation function $G(v, \omega)$ and ensure that the planned path is globally optimal.

$$G'(v, \omega) = \frac{g(n) + (1 - \ln P)h(m)}{G(v, \omega)} \quad (20)$$

$$g(n) = \sum_{(x_n, y_n) - (x_s, y_s)}^{(x_g, y_g)} \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2} \quad ((x_0, y_0) = (x_s, y_s)) \quad (21)$$

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \quad (22)$$

where, $g(n)$ is the movement cost from the starting node (x_s, y_s) to the current node (x_n, y_n) , and $h(n)$ is the estimated cost from the current node (x_n, y_n) to the target node (x_g, y_g) .

There are many redundant sections in the path planned by the improved A* algorithm. These redundant sections not only increase the total length of the path, but also may affect the driving efficiency and energy consumption of the robot. To solve this problem, a key point extraction algorithm is introduced. The core idea of this algorithm is to identify and extract the key turning points from the planned path. These turning points are usually the ones with large direction changes or dense obstacle distribution in the path. By extracting these key points, redundant sections in the path can be eliminated, and a more concise, efficient path with fewer inflection points can be obtained. The specific implementation is as follows:

First, all nodes $\{x_1, x_2, \dots, x_n\}$ in the path are obtained. x_1 is the starting point, x_n is the target node, and V is the key speed sampling set. Then, starting from x_1 , connect $x_2, x_3, \dots$ in a straight line until there is an obstacle in the connecting line with x_m , then add x_{m-1} to V . The nodes from x_1 to x_{m-1} are redundant nodes. At this time, set x_{m-1} as the starting point and connect $x_m, x_{m+1}, \dots$ in a straight line in turn, until there is a straight line connection between the starting point and the target point. Last, the target node is added to V , and all nodes in V are connected in a straight line in turn to complete the optimization.

4.3 Smooth path

In actual path planning should meet the following requirements: the path is continuous, the path curvature is continuous, and the path is easy to follow by the mobile robot. In the grid environment, although the path screened by the improved algorithm has achieved remarkable results in reducing the number of paths turns and turning angles, the planned path still shows the characteristics of an unsmooth, that is, the path curvature is discontinuous. Therefore, to ensure that the mobile robot can travel along the planned path more smoothly, it is particularly important to

smooth the filtered path. Among many smoothing methods, the Bezier curve is favoured for its excellent smoothing performance and flexibility.

The Bezier curve can flexibly control the trend and smoothness of the curve by adjusting the position and number of these control points. The filtered path key points are used as the control points of the Bezier curve and connected to form the control polygon, and then a smooth new path is generated by using the generation algorithm of the Bezier curve. This effectively solves the problem of discontinuous path curvature, makes the planned path more in line with the motion characteristics of the mobile robot, and improves the driving stability of the robot.

For $n + 1$ control point $P_1, P_2, \dots, P_n$, the Bezier curve is:

$$P(t) = \sum_{i=0}^n B_{i,n}(t) \cdot P_i, t \in [0,1] \quad (23)$$

where, P_i is the control point of Bezier curve, t is the independent parameter, and $B_{i,n}(t)$ is the Bernstein basis function of degree n , and it meets the following requirements:

$$B_{i,n}(t) = \sum_{i=0}^n C_n^i \cdot (1-t)^{n-1} \cdot t^i, i = 0,1,2,\dots,n \quad (24)$$

According to the definition of the Bezier curve, P_0 and P_n are the start and end points of the control polygon, respectively. When $n = 1$, the first-order Bezier curve is a straight line with two position points. When $n = 2$, the second-order Bezier curve is a parabola, and there are three position points. When $n \geq 3$, the curve is a higher-order Bezier curve with $n + 1$ position points. To make the path of the Bezier curve fit closer to the path filtered by the improved A* algorithm, the midpoint of each two key path points and the 8 equidistant points close to them are added between each two key path points, and this series of path points is set as the control points of the Bezier curve to generate a smooth path. To sum up, the final optimal obstacle avoidance path planning of mobile robot is achieved.

5. Simulation results and analysis

5.1 Simulation preparation and test indicators

To test the feasibility and effectiveness of local dynamic planning in multi obstacle scenes, that is, whether the mobile robot can avoid obstacles, the proposed algorithm is simulated and tested. Establish a grid map with a side length of 1 m and a size of 50×50 . Set the parameters of the evaluation function as $\alpha = 0.2, \beta = 0.3, \gamma = 0.5$, and the parameters of the improved evaluation function as $\alpha = 0.2, \beta = 0.3, \gamma = 0.15, \lambda = 0.35$. The settings of various speed parameters and time parameters are shown in Table 1. Under the above parameter settings, the global path dynamic obstacle avoidance parameter test software setting interface is shown in Fig. 2.

To provide a comprehensive evaluation of performance and validate the proposed algorithm, the following indicators are defined:

- Path length: it is the key index to evaluate the actual moving distance of the robot from the starting point to the end point. In the field of robot navigation and path planning, the shorter path length shows that the robot can reach the destination more efficiently.
- Obstacle avoidance ability: evaluate the reaction speed and the success rate of obstacle avoidance when the robot encounters obstacles. The reaction speed can be measured by time. The shorter the reaction time, the better the obstacle avoidance capability. The calculation formula for the success rate of obstacle avoidance is as follows.

$$Z_k = \frac{Z_1}{Z} \times 100 \% \quad (25)$$

- Path smoothness: evaluate whether the planned path is smooth and whether there are too many transitions and jitters. The smooth path can reduce the energy consumption and mechanical wear of the robot, and improve the stability and comfort of the movement. The calculation formula is as follows:

$$S = 1 / \sum |L_i - L_{i-1}| / \sum L_i \quad (26)$$

where, L_i is the length of the i -th segment in the path. The higher the path smoothness result, the smoother the path.

Table 1 simulation parameter settings

Parameter name	Numerical value	Parameter name	Numerical value
v_{max}	2 m/s	v_{min}	0
w_{max}	0.7 rad/s	w_{min}	-0.7 rad/s
$\dot{v}_{max}$	0.4 m/s ²	$\dot{v}_{min}$	0
$\dot{w}_{max}$	1.7 rad/s ²	$\dot{w}_{min}$	-1.7 rad/s ²
prediction time T	3 s	time interval Δt	0.1 s



Fig. 2 Global path dynamic obstacle avoidance parameter test software setting interface

5.2 Effectiveness result analysis

To verify the effectiveness of the proposed algorithm, the improved A* algorithm and the algorithm fused with key point extraction are respectively used for obstacle avoidance path planning, and the results are shown in Fig. 3.

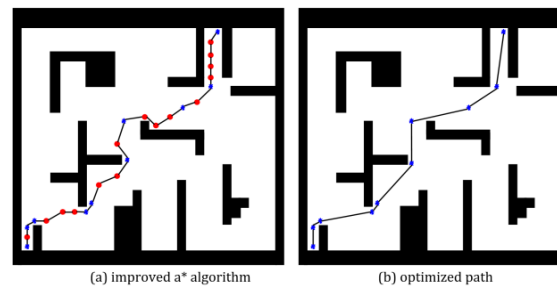


Fig. 3 Obstacle avoidance path planning results

According to the results in Fig. 3, the corresponding distance results can be obtained through the test software. After optimization, compared with the path planned by the improved A* algorithm, the results of the proposed fusion algorithm have been significantly improved. Specifically, the path length is reduced from 39.2536 to 36.9805, which is 5.79 %. This shows that the robot can reach the destination faster and improve the overall execution efficiency. Moreover, the optimized path also significantly reduces the number of nodes. The original 25 nodes have been reduced to only 10 nodes, and the optimization range is as high as 60 %. This not only reduces the complexity of the algorithm, but also helps to improve the stability and reliability of the robot in actual operation. In terms of path inflection points, the number of inflection points was effectively reduced from the original 17 to 8, with an optimization rate as high as 52.94 %. This

shows that the path becomes smoother with the proposed hybrid algorithm, and the robot can reduce unnecessary steering and pauses during driving, further improving the driving efficiency.

To sum up, the results show that the improved A* algorithm and the key point extraction algorithm are better than the improved A* algorithm, which can successfully shorten the path length, reduce the number of nodes and inflection points, thus achieving a significant improvement in robot path planning. This shows that the proposed method is effective and can realize the optimal obstacle avoidance path planning of an intelligent mobile robot.

5.3 Path length result analysis

Based on the above, 10 different 50×50 grid maps are randomly simulated, and the MGWO algorithm in literature [12] and the hybrid algorithm in literature [13] are used as comparison algorithms to test the path length index with the proposed algorithm. The results are shown in Table 2.

According to the results obtained in Table 2, the proposed algorithm is used for obstacle avoidance path planning of 10 scenes, and the path length of each scene after planning is lower than that of another comparison algorithm. The shortest path length after planning using the MGWO algorithm in the literature [12] is 39.658 m, the shortest path length after planning using the hybrid algorithm in the literature [13] is 38.976 m, and the path lengths after planning using the proposed algorithm are less than 37 m. Therefore, using the proposed algorithm for the obstacle avoidance path planning, the path lengths are the shortest, and it can arrive at the target location more quickly and with higher efficiency.

Table 2 Path length results

Scene map number	Path length of the proposed algorithm (m)	MGWO algorithm path length (m)	Hybrid algorithm path length (m)
1	35.765	40.657	39.864
2	35.981	40.879	39.958
3	34.867	39.658	38.976
4	36.724	42.784	40.659
5	35.241	40.823	39.463
6	35.547	40.985	39.769
7	36.837	42.859	40.858
8	35.531	40.991	39.713
9	36.794	41.759	40.957
10	35.785	41.348	39.897

5.4 Analysis of obstacle avoidance results

Based on the above test scenario, the proposed algorithm, the MGWO algorithm in literature [12], and the hybrid algorithm in literature [13] are used to test the obstacle avoidance ability index. The results are shown in Fig. 4.

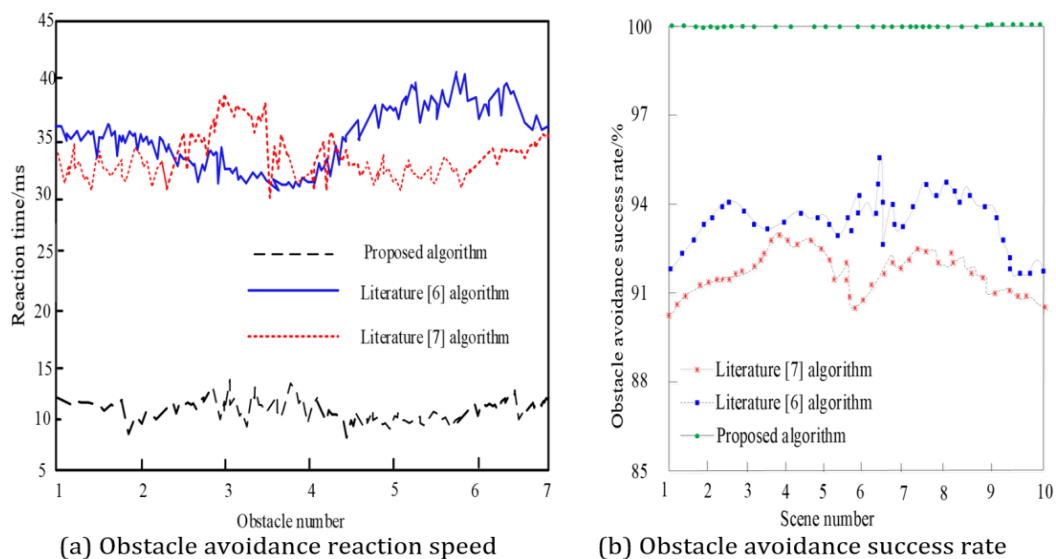


Fig. 4 obstacle avoidance results

According to the results obtained in Fig. 4(a), when the above three algorithms are used for obstacle avoidance path planning, the obstacle avoidance reaction speed is less than 1s, but the comparison of the three results shows that the obstacle avoidance reaction speed of the proposed algorithm is lower than that of the two literature comparison algorithms. When the proposed algorithm is used for obstacle avoidance path planning, its obstacle avoidance response speed can always be maintained within 15 ms. The MGWO algorithm in literature [12] has the lowest obstacle avoidance reaction speed of 32 ms, and the hybrid algorithm in literature [13] has the lowest obstacle avoidance reaction speed of 30 ms. After comparing the three algorithms, the proposed algorithm has a lower obstacle avoidance reaction speed when facing obstacles. According to the results in Fig. 4(b), using the proposed algorithm for obstacle avoidance path planning, the success rate of obstacle avoidance can reach 100 %, and the effect of obstacle avoidance is better. The highest success rate of obstacle avoidance is 93 % using MGWO algorithm in literature [12] and 95 % using hybrid algorithm in literature [13]. After comparing the success rate of obstacle avoidance, it can be concluded that the proposed algorithm can effectively complete the obstacle avoidance path planning. The above results show that the proposed algorithm has a strong obstacle avoidance capability for obstacle avoidance path planning.

5.5 Analysis of path smoothness results

Based on the above test scenario, the proposed algorithm, the MGWO algorithm in literature [12] and the hybrid algorithm in literature [13] are used to test the path smoothness index. The results are shown in Fig. 5.

According to the results obtained in Fig. 5, after the proposed algorithm is used for obstacle avoidance path planning for 10 scenes, the path smoothness results are higher than 97 %. While using the MGWO algorithm in literature [12], its path smoothness result is up to 88 %, and using the hybrid algorithm in literature [13] its path smoothness result is up to 90 %. Therefore, using the proposed algorithm for planning with high path smoothness can effectively reduce the energy consumption and mechanical wear and tear of the robot, and improve the stability and comfort of the movement.

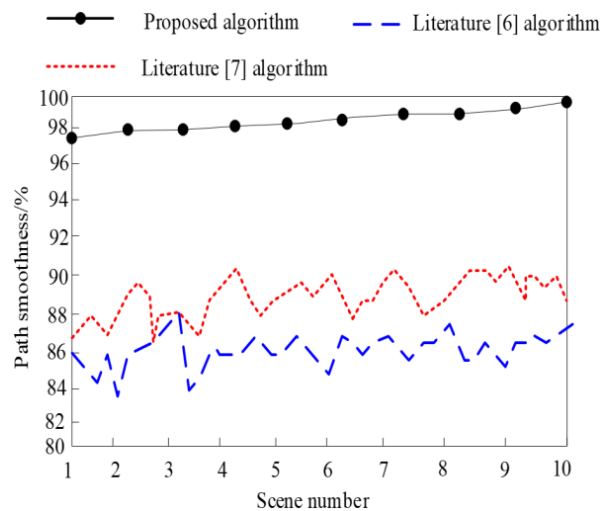


Fig. 5 Path smoothness results

6. Conclusion

With the development of technology, intelligent mobile robots are widely used, but safe and efficient task execution in multi obstacle environments is an urgent problem that needs to be solved in the field of robotics. This study conducted optimal obstacle avoidance path planning for robots in multi obstacle scenarios. The safety boundary condition method and envelope method were used to generate maps and obtain obstacle rates. Adaptive Monte Carlo localization was used, and after velocity sampling and constructing evaluation functions to select trajectories, the

improved A* algorithm was used to introduce obstacle rates and incorporate keypoint extraction algorithms to re plan the path and smooth it out. Although the path planned by this algorithm is short, with an obstacle avoidance response speed of 32 ms, a success rate of 100 %, and a path smoothness of 97 %, it has a driving effect on the development of the robotics field. However, there are potential weaknesses: the construction of maps relies on empirical values, the particle convergence speed of positioning algorithms in special environments may slow down, and the accuracy may be affected. The sampling range and method when constructing trajectories through speed sampling may also affect the results.

Acknowledgement

The authors gratefully acknowledge the support of the National Natural Science Foundation of China (Grant No.: 62001197).

References

- [1] Gültekin, A., Diri, S., Becerikli, Y. (2023). Simplified and smoothed rapidly-exploring random tree algorithm for robot path planning, *Tehnički Vjesnik – Technical Gazette*, Vol. 30, No. 3, 891-898, doi: [10.17559/TV-20221015080721](https://doi.org/10.17559/TV-20221015080721).
- [2] Khaleel, H.Z., Oleiwi, B.K. (2024). Design and implementation low cost smart cleaner mobile robot in complex environment, *Mathematical Modelling of Engineering Problems*, Vol. 11, No. 10, 2869-2877, doi: [10.18280/mmep.111030](https://doi.org/10.18280/mmep.111030).
- [3] Jiang, T., Zhang, S., Wang, R., Wang, S. (2023). Development and verification of an autonomous and controllable mobile robot platform, *Mechatronics and Intelligent Transportation Systems*, Vol. 2, No. 1, 11-19, doi: [10.56578/imits020102](https://doi.org/10.56578/imits020102).
- [4] Muthukumaran, S., Sivaramakrishnan, R. (2019). Optimal path planning for an autonomous mobile robot using dragonfly algorithm, *International Journal of Simulation Modelling*, Vol. 18, No. 3, 397-407, doi: [10.2507/IJSIMM18\(3\)474](https://doi.org/10.2507/IJSIMM18(3)474).
- [5] Khaleel, R.Z., Khaleel, H.Z., Al-Hareeri, A.A.A., Al-Obaidi, A.S.M., Humaidi, A.J. (2024). Improved trajectory planning of mobile robot based on pelican optimization algorithm, *Journal Européen des Systèmes Automatisés*, Vol. 57, No. 4, 1005-1013, doi: [10.18280/jesa.570408](https://doi.org/10.18280/jesa.570408).
- [6] Dabic-Miletic, S. (2024). The challenges of integrating AI and robotics in sustainable WMS to improve supply chain economic resilience, *Journal of Industrial Intelligence*, Vol. 2, No. 2, 119-131, doi: [10.56578/jii020205](https://doi.org/10.56578/jii020205).
- [7] Korabayev, Y., Kosbolov, S., Kubsova, G., Shurenov, M., Duisebayeva, K. (2024). Investigating the design and application of mobile robotic devices with manipulation devices for space technology, *Journal Européen des Systèmes Automatisés*, Vol. 57, No. 2, 345-352, doi: [10.18280/jesa.570204](https://doi.org/10.18280/jesa.570204).
- [8] Jiang, T., Zhang, S., Wang, R., Wang, S. (2023). Development and verification of an autonomous and controllable mobile robot platform, *Mechatronics and Intelligent Transportation Systems*, Vol. 2, No. 1, 11-19, doi: [10.56578/imits020102](https://doi.org/10.56578/imits020102).
- [9] Shi, W., Tang, D.B., Zou, P. (2021). Multi-objective automated guided vehicle scheduling based on MapReduce framework, *Advances in Production Engineering & Management*, Vol. 16, No. 1, 37-46, doi: [10.14743/apem2021.1.383](https://doi.org/10.14743/apem2021.1.383).
- [10] Li, J., Li, J., Fang, H., Jiang, J. (2024). Dynamic energy-efficient path planning for electric vehicles using an enhanced ant colony algorithm, *Tehnički Vjesnik – Technical Gazette*, Vol. 31, No. 2, 434-441, doi: [10.17559/TV-20230510000618](https://doi.org/10.17559/TV-20230510000618).
- [11] Chen, J., Tan, C., Mo, R., Zhang, H., Cai, G., Li, H. (2021). Research on path planning of three-neighbor search A* algorithm combined with artificial potential field, *International Journal of Advanced Robotic Systems*, Vol. 18, No. 3, Article No. 17298814211026449, doi: [10.1177/17298814211026449](https://doi.org/10.1177/17298814211026449).
- [12] Mohanraj, T., Dinesh, T., Guruchandramavli, B., Sanjai, S., Sheshadhri, B. (2023). Mobile robot path planning and obstacle avoidance using hybrid algorithm, *International Journal of Information Technology*, Vol. 15, No. 8, 4481-4490, doi: [10.1007/s41870-023-01475-5](https://doi.org/10.1007/s41870-023-01475-5).
- [13] Chen, Y., Wang, P., Lin, Z., Sun, C. (2023). Global path guided vehicle obstacle avoidance path planning with artificial potential field method, *IET Cyber-Systems and Robotics*, Vol. 5, No. 1, Article No. e12082, doi: [10.1049/csy2.12082](https://doi.org/10.1049/csy2.12082).
- [14] Xia, W., Wang, W., Gao, C. (2022). Trajectory optimization with obstacles avoidance via strong duality equivalent and hp-pseudospectral sequential convex programming, *Optimal Control Applications and Methods*, Vol. 43, No. 2, 566-587, doi: [10.1002/oca.2839](https://doi.org/10.1002/oca.2839).
- [15] Qiu, G., Li, J. (2024). Path planning for unified scheduling of multi-robot based on BSO algorithm, *Journal of Circuits, Systems and Computers*, Vol. 33, No. 7, Article No. 2450133, doi: [10.1142/S0218126624501330](https://doi.org/10.1142/S0218126624501330).
- [16] Jin, X., Wang, Z. (2022). Proximal policy optimization based dynamic path planning algorithm for mobile robots, *Electronics Letters*, Vol. 58, No. 1, 13-15, doi: [10.1049/ell2.12342](https://doi.org/10.1049/ell2.12342).

- [17] Satir, E., Baser, E. (2022). Optimization of interval type-2 fuzzy logic controller using real-coded quantum clonal selection algorithm, *Elektronika Ir Elektrotechnika*, Vol. 28, No. 3, 4-14, [doi: 10.5755/j02.eie.31148](#).
- [18] Chang, C.-K., Elmashtoly, A.M. (2022). Protection coordination index assessment using fuzzy logic controller, *Energies*, Vol. 15, No. 4, Article No. 1377, [doi: 10.3390/en15041377](#).
- [19] Akai, N. (2023). Reliable Monte Carlo localization for mobile robots, *Journal of Field Robotics*, Vol. 40, No. 3, 595-613, [doi: 10.1002/rob.22149](#).
- [20] Jaenal, A., Moreno, F.-A., Gonzalez-Jimenez, J. (2023). Sequential Monte Carlo localization in topometric appearance maps, *The International Journal of Robotics Research*, Vol. 42, No. 13, 1117-1132, [doi: 10.1177/02783649231197723](#).
- [21] Xue, D., Wu, D., Yamashita, A.S., Li, Z. (2023). Proximal policy optimization with reciprocal velocity obstacle based collision avoidance path planning for multi-unmanned surface vehicles, *Ocean Engineering*, Vol. 273, Article No. 114005, [doi: 10.1016/j.oceaneng.2023.114005](#).
- [22] Fang, Z., Liang, X. (2022). Intelligent obstacle avoidance path planning method for picking manipulator combined with artificial potential field method, *Industrial Robot*, Vol. 49, No. 5, 835-850, [doi: 10.1108/IR-09-2021-0194](#).
- [23] Sui, F., Tang, X., Dong, Z., Gan, X., Luo, P., Sun, J. (2023). ACO+PSO+A*: A bi-layer hybrid algorithm for multi-task path planning of an AUV, *Computers & Industrial Engineering*, Vol. 175, Article No. 108905, [doi: 10.1016/j.cie.2022.108905](#).
- [24] Li, C., Huang, X., Ding, J., Song, K., Lu, S. (2022). Global path planning based on a bidirectional alternating search A* algorithm for mobile robots, *Computers & Industrial Engineering*, Vol. 168, Article No. 108123, [doi: 10.1016/j.cie.2022.108123](#).
- [25] Wang, Y.J., Liu, X.Q., Leng, J.Y., Wang, J.J., Meng, Q.N., Zhou, M.J. (2022). Study on scheduling and path planning problems of multi-AGVs based on a heuristic algorithm in intelligent manufacturing workshop, *Advances in Production Engineering & Management*, Vol. 17, No. 4, 505-513, [doi: 10.14743/apem2022.4.452](#).