Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 9, No. 2, 2024

Study of Robot Path Planning with Improved A* And DWA Algorithm Fusion

Jiayi Chen, Josef Cheng*

Xi'an Shiyou University, 710065, Xi'an, Shanxi, China
* Corresponding author:

Abstract: For some problems of the traditional A* algorithm in robot path planning, such as inefficient search, many and complex inflection points in the designed and realized paths, as well as the inability to complete real-time dynamic path planning for obstacle avoidance, this paper proposes an improved A algorithm and combines it with Floyd for path rounding, and finally with DWA (Dynamic Window Approach) algorithm to be Fusion. In the improved A* algorithm, a carefully normalized random obstacle environment is used and the search process is optimized by reducing the search direction and redundant nodes. Meanwhile, the evaluation function is adjusted to achieve the optimal global design solution, and path optimization and obstacle avoidance are performed by the DWA algorithm. The experimental results finally show that when the robot is in a complex environment, the fusion algorithm can ensure the completion of the globally optimal route while avoiding obstacles in a timely and effective manner, which realizes more efficient path planning.

Keywords: Mobile robot; Path planning; Improved A* algorithm; DWA algorithm; Floyd algorithm; Algorithm fusion.

1. Introduction

Mobile robots play a key role in realizing automation and smart manufacturing as the main carrier of objects. In the field of robotics, Automated Guided Vehicles (AGCs), also known as mobile robots [1], are guided by various sensor devices and automatically travel to a designated end point without collision according to a pre-designed and well-planned path. However, the working scenarios of mobile robots in different environments are very complex, and the localization problem is a challenge, and there are usually uncertain human factors, which directly affect the accuracy of the travel route, and thus determine whether the robot can successfully complete the designated navigation task. In order to ensure that mobile robots can successfully complete preset tasks and at the same time satisfy users' needs for localization and navigation accuracy, the accuracy and robustness of mobile robots for path planning in complex environments need to be enhanced.

The navigation process of the robot to reach the target position from the current position can be divided into two parts: global path planning and local path planning. Global path planning refers to planning to get the optimal or near-optimal traveling path in the whole environment, while local path planning is to plan to get the local path that can reasonably avoid obstacles on the basis of following the global path [2-4]. Both global path planning and real-time obstacle avoidance are core problems in mobile robot navigation [5-6].

In order to accomplish the mobile robot can both ensure the global optimal path and have real-time obstacle avoidance ability, this paper proposes a navigation algorithm that integrates the improved A* algorithm and DWA algorithm [13]. The innovations of the algorithm are as follows: 1) optimize the cost function of the traditional A* algorithm to increase its ability to adaptively adjust the attitude; 2) adopt the key point selection method to solve the redundancy problems of the co-linearities of the nodes and the turning points, and only retain the key nodes in the path; 3) construct the evaluation function combining the information of the key points and apply the DWA algorithm, so that the local path planning can be carried out adapting to the global path contour, thus making the the final planned path has higher smoothness and has the ability to avoid obstacles in real time.

2. Improvement of the A* algorithm

2.1. Traditional A* algorithm

The A* algorithm is a commonly used heuristic search algorithm for solving shortest path problems and graph search problems. It finds the optimal path from the starting point to the goal node in a graph where the number of paths is minimized, i.e., the total cost is minimized.

The A* algorithm uses two evaluation functions to select the next node for expansion: the actual path cost (G value) and the heuristically estimated distance (H value). It selects the next expansion node for the current node by adding these two values, i.e., $F(n) = G(n) + H(n)$. Where $G(n)$ represents the actual path cost from the starting point to the current node and $H(n)$ is the heuristically estimated distance from the current node to the target node.

2.2. Optimize search point selection

Traditional A* uses eight directions of search, including up, down, left, right, top left, bottom left, top right and bottom right [7-8]. In this paper, the search direction is reduced to five directions, which reduces the computational cost and storage space, and can effectively improve the search direction is not limited too much. The angle between the line between the target point and the current point and the vertical positive direction (Y) axis is set to β. When the search is reduced to three directions, the angle between β and the Y axis will be expanded to twice as much as the previous one of about 60°, so as to reduce the number of search nodes and improve the computational efficiency.

2.3. Optimizing search nodes

The following are the specific steps to implement the key point selection strategy:

1. set the set of path nodes P={P1,P2,P3,.... ,Pn}, where P1 denotes the start point of the path and Pn denotes the goal point of the path, and the coordinate information of each node is known.

2. Starting from P1, solve for the vector PP1 and the vector PP2, and calculate the magnitude of the angle between the two vectors

3. Categorize them according to the size of the clamping angle:

a) If the magnitude of the angle is zero, i.e., the angle is a straight line, indicating that the three path nodes P1, P2, and P3 are on the same straight line, then P2 is a redundant co-located node and needs to be eliminated

b) If the size of the angle is not zero, i.e., the three path nodes P1, P2, and P3 are not in the same straight line, continue the process to the next step.

4. Solve the line segment equation L(P1,P3) to represent the line segment from path segment P1 to P3. And determine whether or not there is an obstacle within a preset safe distance of that line segment.

5. Processing based on judgment:

a) If there is no obstacle within a safe distance, P2 is a redundant turning point, which is considered to be on a straight line segment and does not affect path planning, and therefore needs to be eliminated.

b) If there is an obstacle within a safe distance, P2 is a necessary turning point and is considered to be a critical point in path planning and needs to be retained.

Through the above steps, the critical points in the path can be filtered and eliminated, removing redundant co-linear nodes and turning points, and retaining the necessary turning points, in order to make the path more streamlined and in line with the actual navigation and planning needs.

2.4. Optimizing the cost function

In the A* algorithm, the cost function determines the path selection during the search process. The A* algorithm can be improved by optimizing the cost function [14]. First, Heuristic function (Heuristic function) In A* algorithm, heuristic function is used to estimate the cost from the current node to the target node, i.e., the residual cost. Second, cost function weight tuning: the cost function in A* algorithm usually consists of two parts, i.e., the cost from the start node to the current node (G(n)) and the estimated cost from the current node to the target node (H(n)). By adjusting the weights of the cost function, the effects of actual and estimated costs on path selection can be balanced. When the weight is 1, the A* algorithm is more inclined to choose the path with smaller actual cost; when the weight is less than 1, it is more inclined to choose the path with smaller estimated cost [15]. In the actual movement process, the weight coefficient of the angle between the actual forward direction and the planning direction determines its traveling direction at the next moment of planning detection; the weight of the obstacles determines its real effective distance traveled in planning the optimal path; the weight of the linear velocity value affects the whole traveling time length of this path planning; and the last but not least, the number and the position of random obstacles, which affects the most crucial global planning duration and distance. Based on each of the above influencing factors, this paper improves its cost function as:

$$F(n)=G(n)+\left|\frac{d+r}{R}\right|H(n)$$

where F(n) is the integrated cost; G(n) is the actual cost from the starting point to the target point; H(n) is the estimated cost from the current point to the target point. r is the distance from the current point to the target point. d is the distance from the starting point to the target point, and R is the distance from the starting point to the destination.

2.5. Floyd's algorithm

Floyd's algorithm is an algorithm for solving the shortest path between pairs of points. It uses the idea of dynamic programming to finally get the shortest path between pairs of points by constantly updating the intermediate nodes in the path.

The core of Floyd's algorithm is the triple loop. The outer loop is used to traverse all possible intermediate nodes, the middle loop is used to traverse all possible start nodes, and the inner loop is used to traverse all possible end nodes. In each loop, the shortest path is updated by comparing the length of the current path through the intermediate nodes with the length of the path without intermediate nodes

3. DWA Algorithm

DWA algorithm (Dynamic Window Approach) is a fast path planning algorithm for robot navigation and mobile intelligences [13]. Its core idea is to select the optimal motion strategy in the current state space of the robot by defining the dynamic window, in order to effectively avoid obstacles and realize the fast generation of the planned path. The basic algorithm idea is: firstly, to obtain the current state of the robot, such as: position, velocity, direction, etc.; secondly, to define the dynamic window, to determine its angular and linear velocity ranges to compose its maximum movable range, called the dynamic window; and subsequently, to generate the optimal obstacle avoidance motion trajectory [8-10].

3.1. Modeling robot motion

DWA performs trajectory simulation by means of the robot's motion model to obtain the optimal path in the simulated motion trajectory. Let v(t) be the linear velocity of the robot at the moment t and w(t) be the angular velocity at the moment t. The positional attitude increment is:

$$\begin{cases}\Delta x=v_t\Delta t\cos(\theta_t)\\ \Delta y=v_t\Delta t\sin(\theta_t)\\ \Delta\theta=\omega\Delta t\end{cases}$$

The bit position at the moment t+1 can be expressed as:

$$\begin{cases}\mathrm{x}(t+1)=x(t)+v_t\Delta t\cos(\theta_t)\\ \mathrm{y}(t+1)=y(t)+v_t\Delta t\sin(\theta_t)\\ \theta(t+1)=\theta(t)+\omega\Delta t\end{cases}$$

At this point, the range of variation of the linear velocity v(t) and angular velocity w(t) is jointly determined by the distance of the nearest obstacle around it and the respective

dynamic maximum velocity.

3.2. Velocity Sampling

Multiple velocity pairs (v,w) can be sampled for reference during robot motion, but in actual sampling, velocity sampling is restrained to a certain range due to the robot's own motor performance and braking distance limitations.

Constraints on the robot's own posture:

$$V_{\mathrm{m}} = \left\{ (v,\omega) \mid v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}] \right\}$$

Where: Vmin and Vmax are both the minimum and maximum linear velocities during robot motion, and Wmin and Wmax are the minimum and maximum angular velocities during robot motion.

Motor acceleration and deceleration constraints:

$$V_d = \left\{ (v,\omega) \mid v \in [v_c - \dot{v}_b dt, v_c + \dot{v}_a dt], \omega \in [\omega_c - \dot{\omega}_b dt, \omega_c + \dot{\omega}_a dt] \right\}$$

Where: Vc and Wc are the current linear and angular velocities; Va is the maximum linear acceleration and Wa is the maximum angular acceleration; Vb is the maximum linear deceleration and Wb is the maximum angular deceleration.

Emergency braking displacement constraints:

$$v_a = \left\{ (v,\omega) \mid v \le \left(2dist(v,\omega)v_b\right)^{1/2}, \omega \le \left(2dist(v,\omega)\omega_b\right)^{1/2} \right\}$$

When the robot movement process suddenly appeared with the obstacle, in order to ensure the safety of the robot itself under the premise and the maximum deceleration conditions, it must stop before hitting the obstacle. In the above equation: dist(v,w) models the distance to the nearest obstacle in the motion trajectory.

The above three cases constrain the range of the robot's moving speeds and provide the basis for the subsequent dynamic window. Let the set of linear velocities of the robot be V_r , then the set of velocities of the robot is:

$$v_r = v_m \bigcap v_d \bigcap v_a$$

3.3. Improvement of the evaluation function

The critical problem in the traditional WDA algorithm is that the dynamic window during the global simulation process using DWA in combination with global simulation can easily fall into the local optimal situation. In order to solve this critical problem, the optimal paths are filtered by an improved evaluation function after simulating multiple global trajectories based on the range of motion speeds and robot model settings. The improved evaluation function equation is as follows:

$$E(v,\omega) = \alpha PHead(v,\omega) + \beta Dist(v,\omega) + \gamma Vel(v,\omega)$$

Where: PHead(v,w) is the angular difference between the end direction of the trajectory and the target direction of the current node; dist(v,w) is the distance between the trajectory and the nearest obstacle; vel(v,w) is the evaluation function of the current speed; α, β, γ are the weighting coefficients. The planned path improved with this evaluation function is not only able to avoid random obstacles, but also more closely fits the globally optimal path traveling.

4. Improved Fusion Algorithm for A* and DWA

In this paper, we integrate the improved A* algorithm and the dynamic window to make the path planning not only conform to the global optimum but also have the ability of obstacle avoidance with local refinement. The idea of fusion: use the improved A* algorithm to plan the global optimal path, get the groups of nodes in the global optimum, use the DWA dynamic window combined with Folyd to perform local optimal solutions between neighboring pairs of nodes, and finally get the improved global optimal path planning. The flow of the fusion algorithm of the improved A* and DWA is shown in Fig. 1:

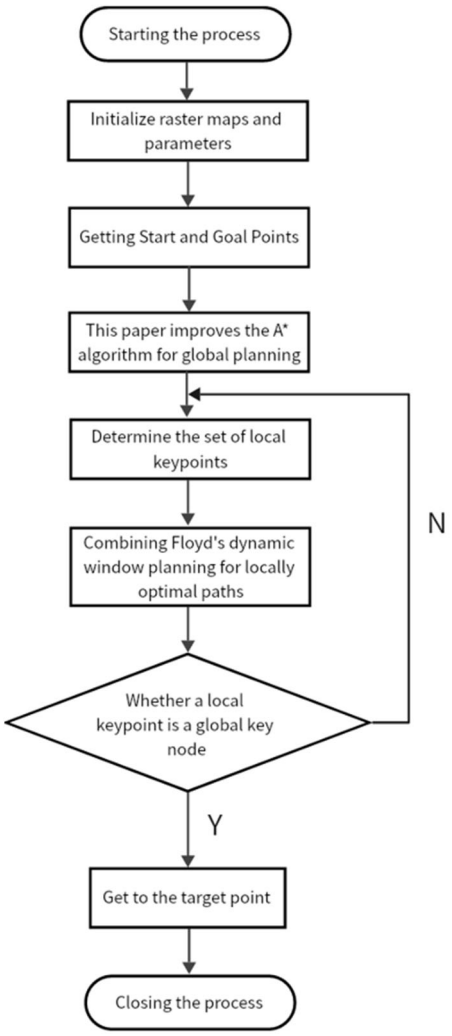


Figure 1. Flowchart of fusion algorithm for improved A* and DWA

5. Simulation Experiment

In this paper, the experimental simulation in different map environments, respectively, compare the node optimization, Folyd optimization and this paper to improve the A* algorithm to optimize the path results, the environments are as follows: Scene 1 is a 25x25 two-dimensional raster map, the starting point of S(26,2), the end point of the T(2,26), the fixed obstacles accounted for nearly 30%; Scene 2 is 30x30 two-dimensional raster map, the starting point of the S(31, 2), ending at T(2,31), with 35% of fixed obstacles; Scene 3 is a

35x35 2D raster map, starting at S(36,2), ending at T(2,36), with 38% of fixed obstacles. Depending on the number of random obstacles added, the performance exhibited by each algorithm is different. First, take scene 1 as an example for comparison: the use of Floyd optimization makes the planned path more rounded and safe, as shown in Figure 2:

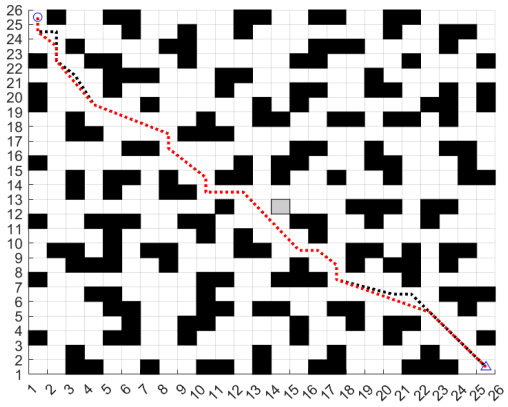


Figure 2. Optimizing results with Floyd

Second, we continue to incorporate node optimization in Scenario 2 to make the whole path planning more explicit, not only increasing the direction angle during the search process, but also eliminating the redundant nodes to achieve a better planning effect, as shown in Fig. 3:

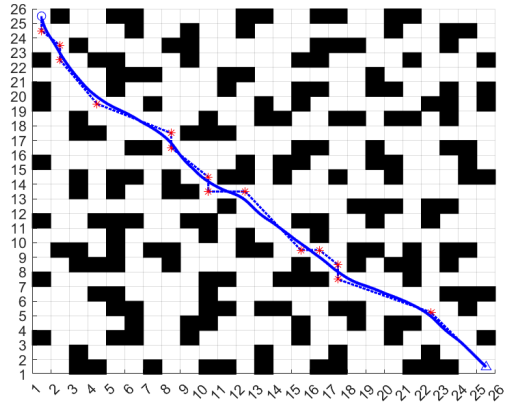


Figure 3. Results of fusion node optimization

Finally, we combine the DWA dynamic window to complete the improvement of the whole algorithm, by improving the node search and cost function of the A* algorithm, together with the fusion of floyd and the DWA dynamic window, the final result of our experiment is shown in Fig. 4:

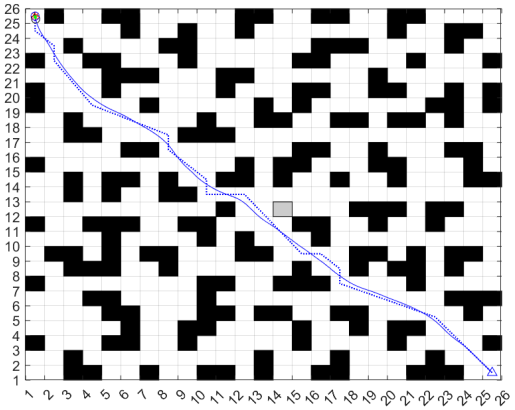


Figure 4. Improved results of this paper

In order to verify the randomness of the experiments, we continued the validation of Scene II and Scene III, also using the same comparison idea for design planning, in the process of driving, according to the different obstacle locations in different scenes, the steering angle range and its speed of obstacle avoidance in each experiment are also different, but it is always kept in a relatively stable value. Angular velocity w(-72, 72)°, linear velocity v(0,0.5)m/s The following node optimization, Folyd optimization and this paper's improved A* algorithm optimization path results for scenario two are shown in Fig. 5, Fig. 6 and Fig. 7, respectively:

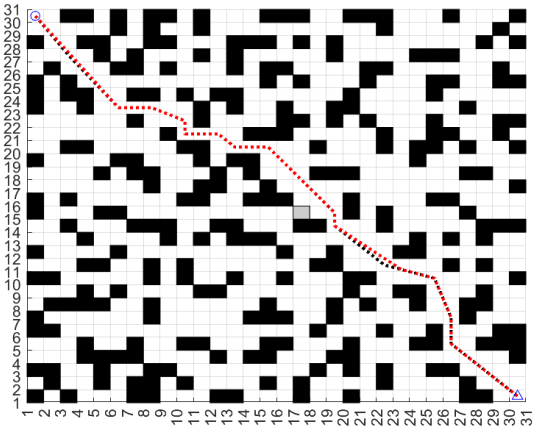


Figure 5. Node optimization results for scenario 2

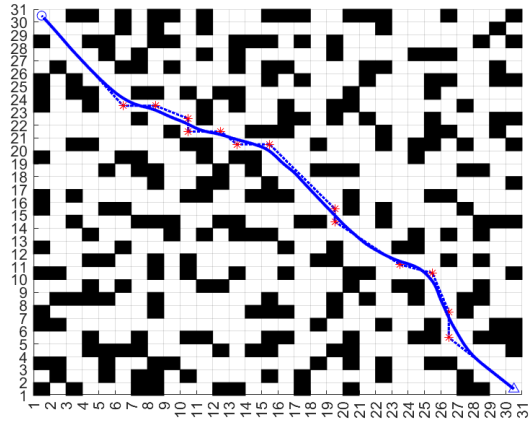


Figure 6. Folyd optimization results for scenario 2

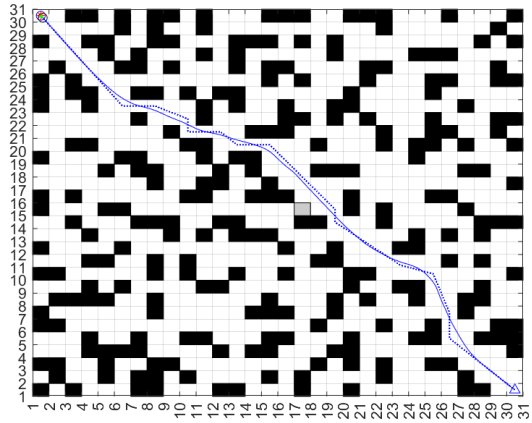


Figure 7. Optimization results of this paper for Scenario 2

The results of the node optimization, Folyd optimization and the improved A* algorithm optimization paths in this paper for scenario 3 are shown in Fig. 8, Fig. 9 and Fig. 10:

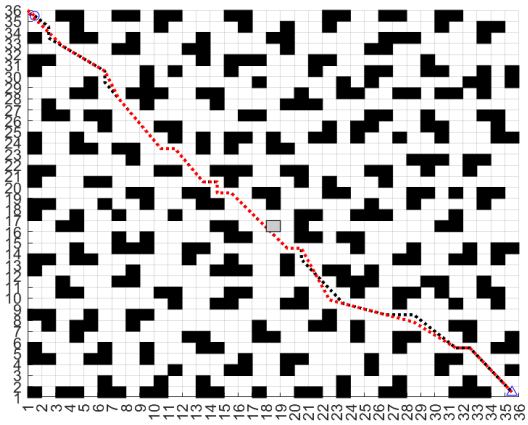


Figure 8. Node optimization results for scenario 3

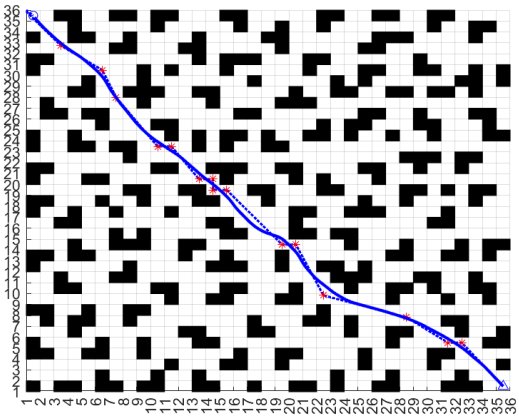


Figure 9. Folyd optimization results for scenario 3

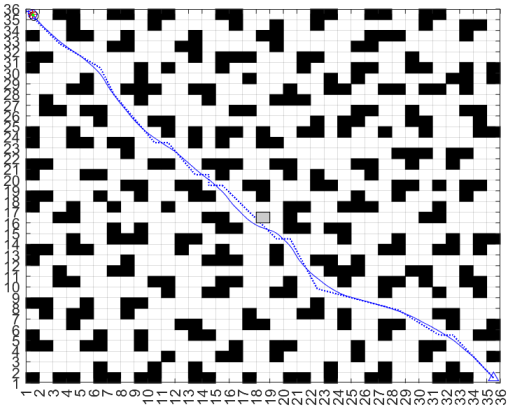


Figure 10. Optimization results of this paper for scenario 3

A comparison of angular and linear velocities is shown in Figure 11:

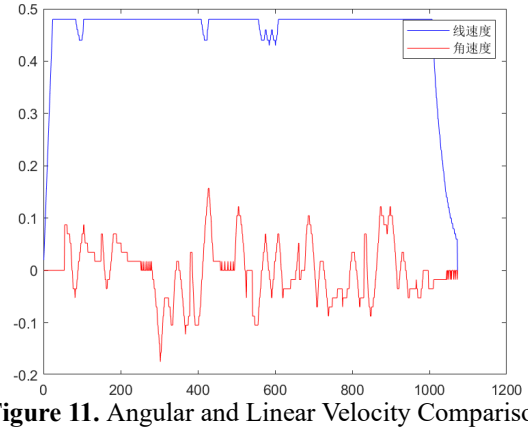
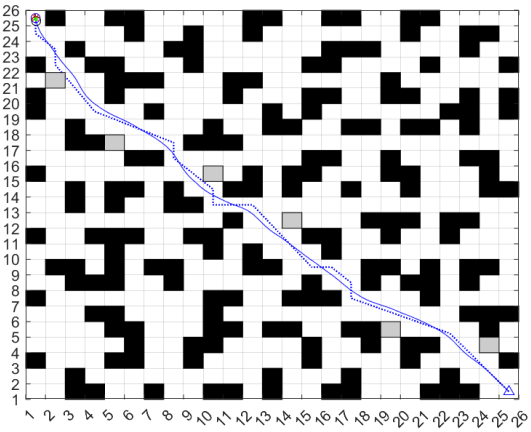
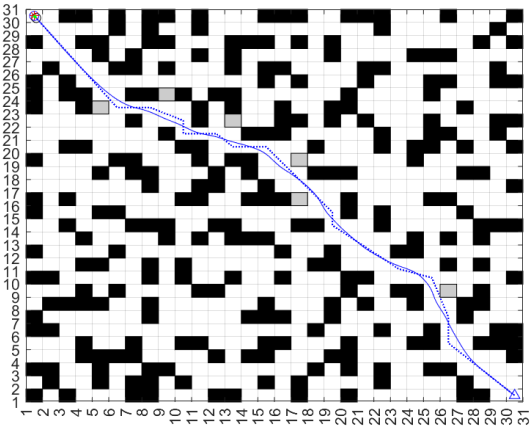


Figure 11. Angular and Linear Velocity Comparison Chart

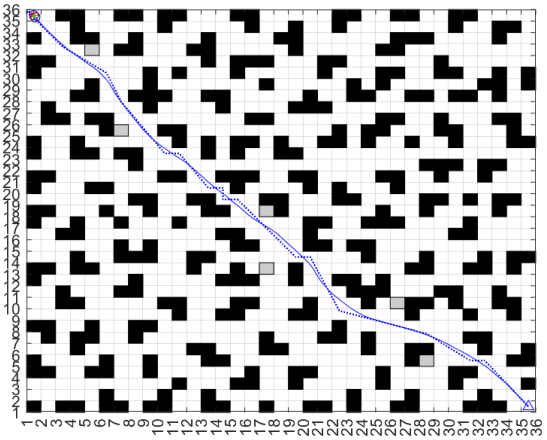
From the above experimental results, it can be seen that the performance of the fusion algorithm of this paper to improve A* is much improved compared to the previous traditional algorithm, and at the same time, when group experiments are conducted, it can be found that with the increasing number of random obstacles at the same time, the improved algorithm of this paper can still be able to carry out random obstacle avoidance. The experimental comparison results are shown in Figures 1-6, 2-6 and 3-6:



Figures 1-6. 6 Random Obstacle Results in Scenario 1



Figures 2-6. 6 Random Obstacle Results in Scenario 2



Figures 3-6. 6 Random Obstacle Results in Scenario 3

Table 1. Planning Path Length Comparison Table

LENGTH \M	SCENE 1	SCENE 2	SCENE 3
RANDOM 1	34.67	42.45	49.10
RANDOM 3	34.67	42.43	49.01
RANDOM 6	34.69	42.40	48.98

Table 2. Planning Path Time Comparison Table

TIME\S	SCENE 1	SCENE 2	SCENE 3
RANDOM 1	420.0	392.8	809.9
RANDOM 3	432.9	396.5	858.8
RANDOM 6	413.4	424.5	831.6

Table 1 and Table 2 show the path planning length and time comparison of the three scenarios with 3 and 6 random obstacles added at the same time. Compared with the above figure with 1 random obstacle, our improved algorithm results in smoother and safer path planning for the relevant requirements. Meanwhile, by comparing the above table, we can see that the length of the planned path is not redundant when random obstacles are added, and the time of the whole planning path is not too long and cumbersome. If applied to real-world scenarios, such as smart driving to avoid "ghost probes", or shopping mall robots in the face of children's playfulness, etc., we can see that the path length is not redundant when random obstacles are added.

6. Summary

The improved A* and DWA dynamic window fusion algorithm proposed in this paper effectively solves the problems of the traditional A* algorithm, such as low efficiency, redundancy and complexity, and inability to adapt to the variability of the environment. The combination of specific refinement to local optimum is achieved on the basis of global planning, and the algorithm makes the robot have reliable real-time obstacle avoidance ability in the real path planning process by optimizing the node search of A* algorithm, deleting redundant nodes and optimizing its cost function. Meanwhile, in practical applications, it provides a novel way of thinking and planning for many areas of path planning, and our future technology and life will generate and face more problems, and more needs and challenges will emerge to be solved. Therefore, deeper explorations are still needed for future technological innovations and researches.

References

[1] Yang W.H. A review of AGV technology development[J]. Logistics Technology and Application,2015,20(11):93-95

[2] JIAOZ,MAK,RONGY,etal.A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs[J].Journal of Computational Science,2018(25):50-57.

[3] Yifan Lin, Yanjie Chen, Bingwei He, et al. Motion planning method for mobile robots without collision detection RT*[J]. Journal of Instrumentation, 2020,41(10):257-267.

[4] DOOPALAM T,BYAMBAA D,DEOK J L.Hybridmotion planning method for autonomous robots usingkinect based sensorfusion and virtual plane approach in dynamic environments [J].Journal ofSensors ,2015(5):1-13.

[5] X. Zhang, S. Cheng, X. Hao, et al. A dynamic path planning algorithm for robots that balances global and local characteristics[J]. Journal of Surveying and Mapping Science and Technology, 2018,(3):315-320.

[6] LI Zhiyin,HUANG Yiquing,XU Yuqiong. Improved variable step size ant colony algorithm for mobile robot path planning[J]. Journal of Electronic Measurement and Instrumentation, 2020, 34(8): 15-21.

[7] Zhao Xiao, Wang Zheng, Huang Chengkan, et al. Mobile robot path planning based on improved A* algorithm[J]. Robotics, 2018, 40(6): 903-910.

[8] WU Yi, OU Mingmin, DUAN Liwei. Research on robot path planning based on improved A* algorithm and dynamic window method[J]. Industrial Control Computer, 2020,33(10) 67-70.

[9] CHENG Legend, HAO Xiangyang, LI Jiansheng, et al. Global dynamic path planning by integrating improved A* algorithm and dynamic window method[J]. Journal of Xi'an Jiaotong University, 2017,51(11):137-143.

[10] Y. X. Wang, Y. Y. Tian, X. Li, et al. Adaptive DWA algorithm for traversing dense obstacles[J]. Control and Decision Making, 2019, 34(5): 927-936.

[11] WANG Zhongyu,ZENG Guohui,HUANG Bo,et al. Improved A~* algorithm for robot global optimal path planning[J]. Computer Applications,2019,39(9):2517-2522.

[12] LAO Cailian,LI Peng,FENG Yu. Path planning for greenhouse robots based on the fusion of improved A~* and DWA algorithms[J]. Journal of Agricultural Machinery, 2021, 52(1): 14-22.

[13] WU Feilong,GUO Shiyong. Fusion of improved A* and dynamic window method for AGV dynamic path planning[J]. Science, Technology and Engineering, 2020,20(30):12452-12459.

[14] Chi X, Li H, Fei JY. Research on random obstacle avoidance method for robots based on the fusion of improved A* algorithm and dynamic window method[J]. Journal of Instrumentation, 2021,42(3):132-140.

[15] Geng H. F., Shen J. J. Improvement and validation of A* algorithm on robot path planning[J]. Computer Applications and Software, 2022,39(1): 282-286.