

Robot Path Planning Algorithm based on Improved A* and DWA

Haisheng Song, Deyang Zhang*

Collage of Physics and Electronic Engineering, Northwest Normal University, Lanzhou, Gansu 730070, China

* Corresponding author: Deyang Zhang (Email: 1361610916@qq.com)

Abstract: Aiming at the problems of low search efficiency of A* algorithm in traditional path planning, many redundant points, and inability to avoid unknown obstacles in real-time in complex environments, this paper proposes a path planning algorithm based on A* combined with Dynamic Window Approach (DWA) algorithm. First, the evaluation function of the traditional A* algorithm and the expansion strategy of sub-nodes are improved to improve the safety and search efficiency of the global path. Then the redundant nodes in the global path are processed to reduce the number of turning points and improve the smoothness of the global path. to improve the instability and energy consumption of the robot during travel; finally, based on the global path planning, the DWA algorithm is introduced to perform path planning in the local unknown environment. The local path planning is completed by retaining the key path turning points as intermediate path guidance. Real-time obstacle avoidance. Through simulation experiments, the effectiveness and feasibility of the fusion algorithm are verified.

Keywords: Path Planning; A* Algorithm; Dynamic Window Approach; Fusion Algorithm.

1. Introduction

With the rapid growth of artificial intelligence and the expansion of the robot industry, robots are widely utilized in manufacturing, medical, logistics, and various other fields[1]. Path planning is a crucial prerequisite for robots to execute tasks[2], and its measurement index is based on the evaluation standard between the starting point and the target point to plan an optimal and collision-free path in the shortest time[3]. Current path planning algorithms can be categorized into global static path planning with a known environment and local path planning with unknown environment information. Commonly used algorithms in global path planning include the A* algorithm [4], artificial intelligence algorithm [5], Dijkstra algorithm [6], etc. Planning algorithms in local path planning include the artificial potential field method [7], dynamic window method (DWA) [8], time elastic band (TEB) [9], etc.

The A* algorithm is widely employed in path planning due to its high search efficiency and ease of implementation. However, the traditional A* algorithm has shortcomings such as many vertices, susceptibility to contact with the edges of obstacles, and uneven paths. Despite DWA's proficiency in avoiding obstacles in unknown environments, it faces challenges in complex environments since it is solely guided by the final goal, making it prone to local optimality and leading to failures in local path planning.

In response to these challenges, this paper proposes a path planning algorithm that integrates the improved A* with the DWA algorithm. It enhances and optimizes the traditional A* algorithm, introducing environmental factors to refine the heuristic function. This allows decisions based on obstacle density, dynamic adjustment of the degree, constrained selection of sub-nodes, avoidance of oblique crossing of obstacles, elimination of redundant points, reduction of path turns, and introduction of a safety factor to enhance the path safety distance. The approach completes global path planning and optimizes it. Building upon this, local path planning is conducted based on potential unknown static obstacles and

dynamic obstacles. Key nodes in global path planning are chosen as a guide for the local path, and an evaluation function with global information is constructed to achieve the fusion of the two algorithms. This results in an enhancement of the completeness, security, and real-time nature of the path.

2. Establishment of Environmental Model

The prerequisite for robot path planning is to select an appropriate environment model. Currently, commonly used modeling methods include grid method, topology method and geometric method, because the grid method has the characteristics of easy implementation, simple programming and consistency in the construction of environmental models. Features. Therefore, this article uses the grid method to construct an environmental model. The environmental space is divided into grids of the same size. According to the environmental information, each grid is represented by blank (representing the drivable area) and black (representing the obstacle area). The grid map is shown in Figure 1.

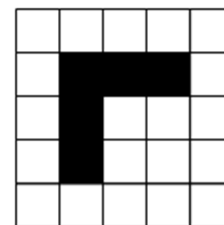


Figure 1. Grid map

3. A* Algorithm Improvements

3.1. Traditional A* Algorithm

The A* algorithm is a classic heuristic search algorithm that combines Dijkstra and breadth-first traversal algorithms. It searches for nodes from the starting point, selects a node with the smallest evaluation function value as the starting point of the next search node, and continues to move toward

the target point. Finally, the shortest distance is found from the starting point to the target point in the global path. Its core functions are:

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

In the formula, n represents the current node, represents the total cost value from the starting point to the target point, represents the actual cost value from the starting point to the current node, represents the estimated cost value from node n to the target point. Among them, Manhattan distance, Euclidean distance and Chebyshev distance are commonly used as heuristic function. This paper selects Euclidean distance as the heuristic function of, and its expression is:

$$h(n) = \sqrt{(G_x - S_x)^2 + (G_y - S_y)^2} \quad (2)$$

Among them, (X_n, X_n) , (G_x, G_y) is the current node coordinates and target point coordinates.

3.2. Heuristic Function Improvement

In the traditional A* algorithm, since $h(n)$ plays a key role, in the early stage of path search, appropriately adjusting the weight of $h(n)$ will help reduce the waste of search resources and achieve path optimization. First, the obstacle rate K is expressed according to the number of obstacles in the grid map environment:

$$K = \frac{O_{bs}}{(G_{x+1} - S_x) \cdot (G_{y+1} - S_y)} \quad (3)$$

In the formula, A is the number of obstacles from the starting point to the target point, b is the coordinate of the starting point, and the obstacle rate K is introduced into the core function, and its expression is:

$$f(n) = g(n) + \left(1 + \frac{1}{1 + e^k}\right)h(n) \quad (4)$$

Dynamically adjust according to the obstacle rate in the map environment to balance the two functions.

3.3. Optimize Child Node Expansion Strategy

The traditional A* algorithm uses 8 neighborhood expansion, as shown by the blue arrow in the parent-child node diagram in Figure 2:

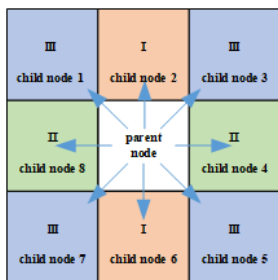


Figure 2. Parent-child node graph

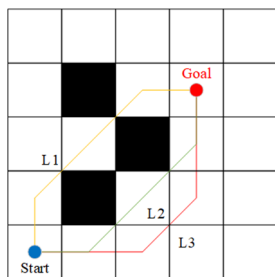


Figure 3. Child node expansion path

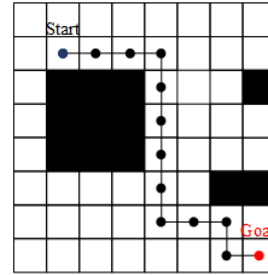
According to the A* algorithm's rules for selecting child

nodes, if the child node is an obstacle, it will cause the node to touch the edge of the obstacle or even cross the two obstacle vertices diagonally, as shown in L1 and L2 in Figure 3.

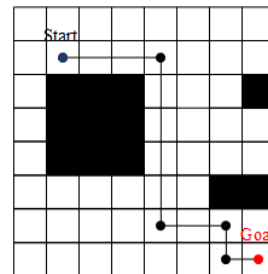
In order to improve the safety of unmanned boats and avoid the above risks, it is necessary to restrict the selection of child nodes. According to the relationship between the parent node and each child node in Figure 2, the positional relationship between the parent node and the child node is divided into three types: If it is a type 1 or 2 situation, the nodes adjacent to the obstacle will not be selected. If it is a type 3 obstacle, no constraint processing will be performed. After the above constraints, collisions with obstacles can be effectively avoided. The improved A* is shown as L3 in Figure 3.

3.4. Redundant Node Handling

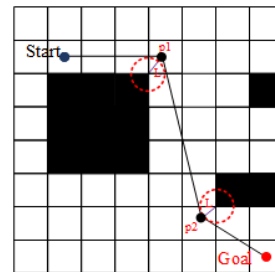
In the traditional A* path planning, there are many collinear nodes and turning points, resulting in a non-optimal path that is not smooth. At the same time, under the influence of many such nodes, the robot will move according to the These redundant nodes constantly change directions, increasing instability and energy consumption during travel. The nodes are shown in Figure 4.



(a)Traditional A* path planning node



(b)Key point extraction



(c)Path after removing redundant points

Figure 4. Remove redundant points

(1) Start from the starting point S and gradually search for the next node until the target point. If several nodes are on the same straight line and there are no obstacles directly on the straight line, then only the starting point and end point of the straight line will be retained, the nodes after screening are shown in Figure 4(b).

(2) Since in A* path planning, the robot is treated as a particle, the actual volume of the robot is ignored, and the safe distance D_{st} to prevent collision with the grid is quoted to form the optimal path, $D_{st} = R_{grid} \cdot \sqrt{2}/2$, where R_{grid} represents each grid. The side length of the grid.

In Figure 4(b), a binary search is performed from the starting point to the target point on the path removing the intermediate nodes. If there are no obstacles, the original path is retained. When an obstacle is encountered, the last selected node is retained. And calculate the distance L from the center to the obstacle. By comparing L with the safe distance D_{st} , if $L > D_{st}$, keep the previous node, if $L \leq D_{st}$, delete the previous node. The final path is shown in Figure 4(c).

It can be seen that redundant points, path length, and number of turns are significantly reduced.

4. DWA Path Planning Algorithm

In the actual working environment, there are unknown static and dynamic obstacles. These factors will have an impact on the robot's path. Therefore, in order to achieve safety during the work process, this paper uses the dynamic window method to perform local planning after the global path planning is completed. Path planning in unknown environments.

4.1. Robot Velocity Sampling

In the actual movement of the robot, there will be a certain range of linear speed and angular speed due to the constraints of its own design and structure. V_m is the set between the minimum and maximum speed of the robot. The speed includes the linear speed and angular speed, expressed as (v, m) , then V_m can be expressed as:

$$V_m = \{(v, m) | v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \quad (5)$$

maneuverability constraints:

$$V_d = \{(v, m) | v \in [v_c - v' \Delta t, v_c + v' \Delta t], [\omega_c - \omega' \Delta t, \omega_c + \omega' \Delta t]\} \quad (6)$$

braking distance constraint:

$$V_a = \{(v, \omega) | v \leq \sqrt{2 \text{dist}(v, \omega) \cdot \dot{v}}, \omega \leq \sqrt{2 \text{dist}(v, \omega) \cdot \dot{\omega}}\} \quad (7)$$

After integrating the above three constraints, we can obtain the sampling space V_r of all feasible speeds. In order to ensure the safety of the unmanned boat, its speed should satisfy all the above constraints. Therefore, V_r is the intersection of the

above three sets, expressed as:

$$V_r = V_m \cap V_d \cap V_a \quad (8)$$

4.2. Improved Trajectory Evaluation Function

After determining the speed sampling space, multiple sets of feasible speed pairs can be discretized from it. After obtaining the estimated trajectories corresponding to multiple sets of speeds, it is necessary to use the evaluation function as a criterion to evaluate and select all trajectories, and finally select the speed corresponding to the optimal trajectory is used as the navigation speed of the next unit moment. In the traditional dynamic window method, the parameters considered in the evaluation function are azimuth, speed, and distance.

$$G(v, w) = \alpha \cdot \text{head}(v, w) + \beta \cdot \text{vel}(v, w) + \gamma \cdot \text{dist}(v, w) \quad (9)$$

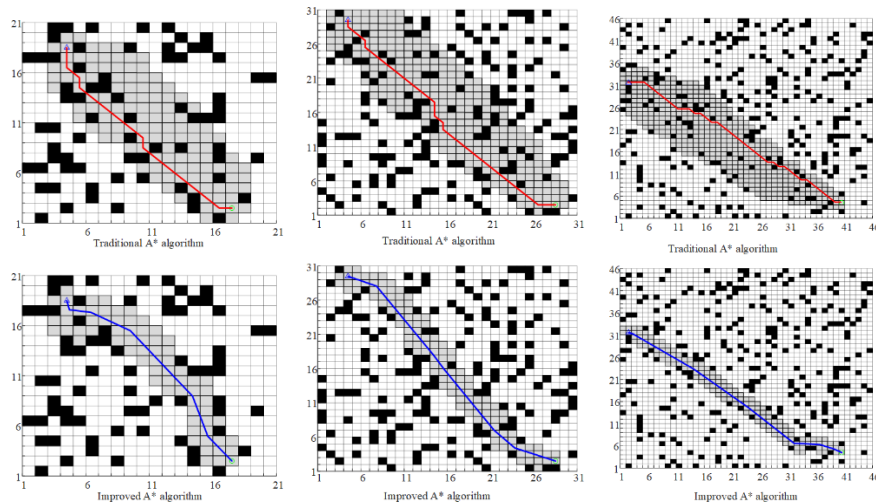
Among them, α , β and γ are the weighted coefficients of three evaluation functions respectively, $\text{head}(v, w)$ is the azimuth angle deviation between the direction trajectory of the current movement direction and the target point; $\text{vel}(v, w)$ evaluates the current speed, and $\text{dist}(v, w)$ is the distance between the current movement trajectory and the obstacle. Distance, analysis of the above evaluation function shows that there is only one final target as a guide in the azimuth angle. If there are too many obstacles in the environment, the path will fall into a local optimum, because this article adopts the key to improve A* global path planning. The turning point serves as a guide point to reach the target point. In addition, for unknown static and dynamic obstacles in the local environment, the distance function must be improved at this time to construct a new evaluation method that combines turning points and unknown obstacles. function, the new evaluation function is:

$$G_{new}(v, w) = \alpha \cdot \text{Chead}(v, w) + \beta \cdot \text{vel}(v, w) + \gamma \cdot \text{dist}(v, w) + \varepsilon \text{dist}_m(v, w) \quad (10)$$

In the formula: $\text{Chead}(v, w)$ is the azimuth angle deviation between the end point of the trajectory and the current key turning point. According to the sequence of turning points, it guides the robot to move to the final target point.

5. Experimental Simulation

5.1. Improved A* Algorithm Simulation Analysis



(a) 20*20 map environment (b) 30*30 map environment (c) 45*45 map environment

Figure 5. Comparison of path simulation in different map environments

In order to verify the accuracy and effectiveness of the global path planning algorithm of the improved A-star algorithm, this article constructed three different map environments of 20*20, 30*30, and 45*45 for simulation testing. The simulation experimental environment is MATLAB 2018b. The blue triangle represents the starting point, the green circle represents the target point, and the gray grid represents the searched grid during the path search process. The experimental results are shown in Figure 5, and the performance indicators of the algorithm are shown in Table 1.

From the simulation results in Figure 5, it can be clearly seen that in each different grid map, the traditional A* algorithm has many turning points, and there are situations where it passes directly through the middle of the obstacle, or collides with the edge. during the search process, a large number of nodes were also expanded. In the improved A* algorithm, the path is smoother than the traditional A*, and at the same time, under the constraint of a safe distance, the risk of collision with obstacles is avoided.

From the data in Table 1, we can see that in the 20*20 map, the path length and the number of turns are roughly the same, but in the path planning process, compared with the traditional A* algorithm, the improved A* reduces the

number of nodes by 42.14%, and the path number is reduced by 42.14%. The security has also been improved. In the 30*30 map environment, the number of nodes is reduced by 56.68%, and the number of time turns is reduced by 3 times. In the 45*45 map environment, with the addition of the obstacle coefficient, the number of nodes is reduced. 65.16%, the number of turns is reduced by 8 times, the path is still safe at this time, and the length of the path is reduced by 3.1%, indicating that the improved A* algorithm still has good adaptability to complex maps.

Table 1. Comparison of path planning simulation results under different map environments

Map type	Algorithm type	Path length/m	Number of nodes/node	Number of turns
20*20	Traditional A*	21.97	121	6
	Improve A*	21.71	70	5
30*30	Traditional A*	38.11	277	8
	Improve A*	37.27	120	5
45*45	Traditional A*	49.18	442	14
	Improve A*	47.66	154	6

5.2. Simulation Experiment of the Fusion Algorithm

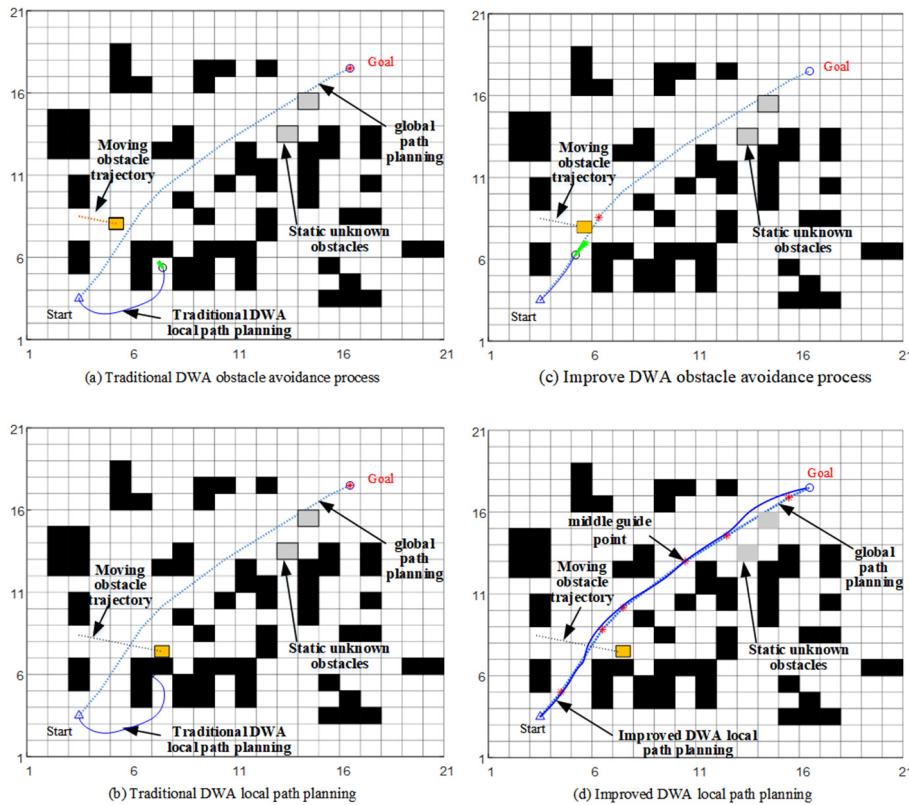


Figure 6. Simulation of two fusion algorithms

In order to verify the adaptability of the hybrid algorithm, a simulation experiment was conducted on a 20*20 map. There is one dynamic obstacle and two static obstacles in the map environment. The orange square represents the dynamic obstacle and the gray represents the static unknown obstacle. START represents the starting point, Goal represents the target point, and the dotted line represents the global path planning of improved A*. The paths of the two fusion algorithms are shown in Figure 6.

In Figure 6a, in the traditional DWA algorithm, the final

target point is regarded as a single target point. In an environment with many obstacles and dynamic obstacles, the final result is shown in Figure 6(b), falling into a local optimal, causing path planning to fail.

In Figure 6b, the red '*' in the global path represents the reserved path turning points. According to the order of turning points, the robot is guided to move to the final target point. By adding the guidance of key nodes, under the fusion algorithm in Figure 6d, the robot's path is effective. It avoids dynamic obstacles and static unknown obstacles, and avoids

falling into local optimality when avoiding dynamic obstacles. Moreover, compared with the global path, the path is smoother, completing path planning.

6. Conclusion

In order to solve the problem that in complex environments, the traditional global A* path planning algorithm has many search nodes, many redundant points, and cannot avoid dynamic obstacles, this paper proposes a path planning that improves A* combined with the DWA algorithm. By adding the DWA algorithm, And the turning point of the A* path is selected as the intermediate node of DWA to realize the fusion of algorithms on the basis of global optimality, so that the robot has the ability to avoid local static and dynamic obstacles in complex environments. Experimental results show that the improved fusion algorithm can effectively avoid local optima and collisions with static and dynamic obstacles. Reaching the end point safely proves the feasibility of the fusion algorithm in robot path planning.

References

- [1] Memetimin H L M Q Q L Y L .Path Planning Technique for Mobile Robots: A Review[J].Machines,2023,11(10).
- [2] Jicun Y W G L G X .Mobile robot path planning with reformativebatalgorithm.[J].PloS one,2022,17(11):e0276577-e0276577.
- [3] Behnam A M .A new path following scheme: safe distance from obstacles, smooth path, multi-robots[J].Journal of Ambient Intelligence and Humanized Computing, 2023, 14(4): 4621-4633.
- [4] Qing R J E M K .A*-Based Co-Evolutionary Approach for Multi-Robot Path Planning with Collision Avoidance[J]. Cybernetics and Systems,2023,54(3):339-354.
- [5] Shuo W .Artificial Intelligence Path Planning for Marine Environment[J].Journal of Physics: Conference Series, 2021, 1852 (2).
- [6] Tan Z X Z P .A Heuristic Integrated Scheduling Algorithm Based on Improved Dijkstra Algorithm[J]. Electronics, 2023, 12 (20).
- [7] Chen Z H L H G .UAV Path Planning Based on Improved Artificial Potential Field Method[J].Aerospace,2023,10(6).
- [8] Sheng G Z J Y L .A Robot Path Planning Method Based on Improved Genetic Algorithm and Improved Dynamic Window Approach[J].Sustainability,2023,15(5):4656-4656.
- [9] Chenyang S B S X S .A Solution for Trajectory Planning and Control of Cooperative Steering Mobile Robot Based on Time Elastic Band[J].Journal of Computer and Systems Sciences International,2022,61(6):1046-1057.