10.18686/aitr.v2i3.4408

Robot Path Planning Based on Improved A* Algorithm and Artificial Potential Field Method

Xiancheng Fan, Xinyu Ling, Hongbin Huang

School of Electrical and Electronic Engineering, Anhui Institute of Information Technology, Wuhu 241000, China.

Abstract: In response to the problems of long planning paths, large turning angles, and inability to avoid dynamic obstacles in traditional A* algorithm robot path planning, this paper proposes a path planning algorithm that combines improved A* algorithm with artificial potential field method. Firstly, the improved A* algorithm searches the neighborhood and heuristic function. Compared with some algorithms, the improved A* algorithm reduces the optimal path distance, search nodes, simulation time, and turning angles by 22.78%, 80.65%, 69.84%, and 50% respectively. The improved A* algorithm is further optimized by removing redundant nodes and smoothing the path, reducing the optimal path, simulation time, and turning angles by 2.08%, 9.1%, and 36.36% respectively compared to the first optimization. For local path planning using artificial potential field, the artificial potential field function and adaptive step size are improved. Simulation results show that the improved algorithm can overcome the problems of local minima and unreachable targets. Finally, the integrated algorithm simulation shows that it can solve the problem of A* algorithm's inability to avoid dynamic obstacles and guide the robot to move along the optimal path.

Keywords: A* algorithm; Robot; Path planning; Artificial potential field; Dynamic obstacles

Introduction

With the explosive development of artificial intelligence, intelligent robots have been applied in various industries, including industrial, medical, and service industries. In robot navigation, path planning is a crucial research direction for robot motion. Intelligent path planning algorithms can generate an optimal path from the starting point to the destination, ensuring safe, precise, and collision-free robot movement.

Path planning algorithms can be divided into global path planning algorithms and local path planning algorithms. Global path planning algorithms include Dijkstra's algorithm [1], A* algorithm [2], ant colony algorithm [3], genetic algorithm [4], particle swarm optimization algorithm [5], and others. Local path planning includes artificial potential field method [6], DWA algorithm [7], and others. Dijkstra's algorithm has a high time complexity and is not suitable for robot path planning. Intelligent algorithms such as ant colony algorithm and genetic algorithm have disadvantages such as high computational complexity and slow convergence speed. The A* algorithm, with the introduction of a heuristic function, can solve the problem of redundant nodes in path planning. However, the A* algorithm can only optimize global paths in static maps and cannot accurately avoid obstacles in unknown and dynamic environments. Local path planning can avoid obstacles in unknown and dynamic environments, but it cannot find the globally optimal path.

To address the aforementioned issues, this paper proposes a robot path planning algorithm that integrates improved A* algorithm and improved artificial potential field method. The algorithm improves the neighborhood search method and the heuristic function of the A* algorithm and removes redundant nodes. For the artificial potential field method, the algorithm improves the potential field function and adjusts the step size to escape obstacle influence. To overcome the limitation of A* algorithm in unknown and dynamic environments, an improved local path planning algorithm based on artificial potential field is introduced. It can compensate for the shortcomings of the A* algorithm and achieve optimal path planning in complex environments.

1. A* Algorithm

1.1 Environmental Model

Currently, common methods for constructing environment maps for robot path planning include the grid method, visibility graph method, and vector method, among which the grid method is the most widely used[8]. The grid method constructs the robot's movement space into a uniformly distributed grid space, as shown in Figure 1 below. In the figure, black represents obstacle areas, and white represents free walking areas. The path planning algorithm aims to find a path from the start point to the end point within the grid space to complete the robot's path planning.

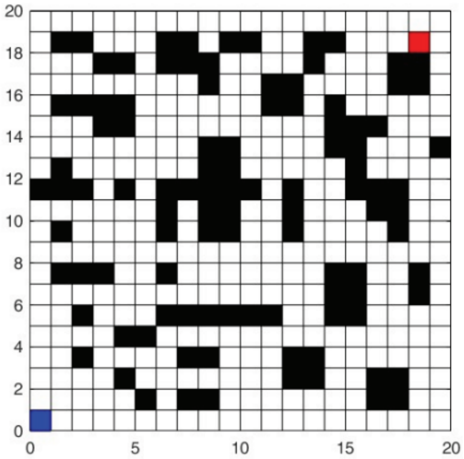


Figure 1: Map Model

1.2 The Classic A* Algorithm

The A* algorithm is a heuristic pathfinding algorithm based on Dijkstra's algorithm, which improves the heuristic function to enhance performance [9]. The grid evaluation function of the A* algorithm is shown in Equation 1:

$$f(n) = g(n) + h(n) \tag{1}$$

where (f(n)) represents the total cost of the nth grid, (g(n)) is the actual cost from the initial grid to the nth grid, and (h(n)) is the estimated cost from the nth grid to the target grid. The choice of the heuristic estimate (h(n)) is crucial:

1. When (h(n)) is less than the actual distance, the number of search nodes increases, expanding the search range and reducing efficiency, but the algorithm can find the optimal path.

2. When (h(n)) equals the actual distance, search efficiency is highest.

3. When (h(n)) exceeds the actual distance, the number of search nodes decreases, the search range narrows, and efficiency increases, but the algorithm may not find the optimal path.

4. When (h(n)) equals zero, the A* algorithm degrades to Dijkstra's algorithm, greatly increasing search time and the number of nodes, but it guarantees finding the optimal path.

Common methods for calculating (h(n)) include Manhattan distance (h_M), Euclidean distance (h_E), and Chebyshev distance (h_C) [10]. The distance formulas are shown in Equations 2, 3, and 4, respectively, where (x_C) and (y_C) are the coordinates of the robot's current position, and (x_g) and (y_g) are the coordinates of the robot's target position. This paper uses the most common Euclidean distance as the basic heuristic function for the A* algorithm.

$$h_M = \left|x_c - x_g\right| + \left|y_c - y_g\right| \tag{2}$$

$$h_E = \sqrt{\left(x_c - x_g\right)^2 + \left(y_c - y_g\right)^2} \tag{3}$$

$$\begin{aligned} &h_C = dx + dy + \left(\sqrt{2} - 2\right) * \min(dx, dy); \\ &dx = \left|x_c - x_g\right|, dy = \left|y_c - y_g\right| \end{aligned} \tag{4}$$

Robots walk in grid maps, spreading from the starting point to adjacent grid nodes. The classic A* algorithm uses four directions for diffusion, also known as four-neighborhood search. After continuous improvement by researchers, the original four-neighborhood search method has been increased to 8-neighborhood, 16-neighborhood and other search methods [12].

1.3 Improving the A* Algorithm

The classic A* algorithm is used for global path planning and can effectively find the global path, but it has several shortcomings such as redundant nodes, long search time, and excessive turns. Sometimes it cannot even find the optimal path, which seriously affects the efficiency of robot path planning [11]. This paper optimizes the A* algorithm from the following aspects.

1.3.1 Optimization of searching neighborhood

The classic A* algorithm usually performs four-neighborhood search, eight-neighborhood search, 16-neighborhood search, etc. Due to the single direction of the four-neighborhood search method, it ignores the possibility of the optimal path that the robot may take diagonally; the 16-neighborhood search method significantly increases the number of search nodes, increases the complexity of the algorithm, and there is a possibility of moving through obstacles [12]. In view of the above problems, this paper chooses variable neighborhood search method for neighborhood optimization. Through the robot's current node position and the target point for the connection and the positive direction of the angle α, through the size of the angle α to dynamically allocate the search domain.

1.3.2 Optimization of heuristic function

The total cost searched by the A* algorithm is judged by the function f(n) for each searchable node, so the f(n) function is the most direct factor affecting the search ability of the A* algorithm [13]. To optimize the A* algorithm, a reasonable heuristic function is necessary. Firstly, this paper uses the most original heuristic function as the starting function of the A* algorithm, and modifies the weights of the actual cost and estimated cost by adding a weighting factor to the actual cost and estimated cost. The formula is shown in Equation 5.

$$f(n) = \alpha g(n) + \beta h(n) \tag{5}$$

Based on the distance between the robot's current position and the target position, the estimation function should be adjusted. As shown in Formula 6.

$$\mathrm{S} = \sqrt{\left(x_{\mathrm{C}} - x_{C}\right)^{2} + \left(y_{0} - y_{g}\right)^{2}} \tag{6}$$

$$w_n = \gamma * \exp(S) * (1 + h_E / S) \tag{7}$$

Where x_C, y_C represent the starting point coordinates, while x_g and y_g represent the ending point coordinates. exp stands for the exponential function, used to calculate powers of e. γ is the angle correction coefficient.

The improved total cost function is:

$$f(n) = \alpha * g(n) + \beta * h(n) * w_n \tag{8}$$

1.3.3 Optimization of redundant nodes

The classic A* algorithm is composed of continuous search nodes, which inevitably leads to redundant and unnecessary nodes, increasing the number of turns for the robot and affecting its running performance. Therefore, it is necessary to delete redundant nodes. The optimization of redundant nodes is divided into two steps:

1) Traverse the search nodes in the A* algorithm, select three nodes in turn, and connect the first node with the third node. Judge the distance between the connection line and the obstacle. If the distance between the connection line and the obstacle is less than the set safety distance, it means that the path planning does not meet the requirements and the second node is retained. When the distance between the connection line and the obstacle is greater than the set safety distance, it indicates that the route planning is successful and the intermediate node is deleted.

2) Repeat the process until all nodes are traversed, retain the remaining nodes, and generate the path.

2. Artificial Potential Field Method

2.1 Classical Artificial Potential Field Method

The Artificial Potential Field (APF) method is widely used as a local path planning algorithm for robots. It was first proposed by Khatib.O. In the APF algorithm, the target point and obstacles are virtually generated attractive and repulsive forces for the robot's movement, guiding the robot's motion through the combined action of gravity and repulsion[13]. The repulsive force field of obstacles is shown in Figure 2 (F_{rep}), and the attractive force field of the target point is shown in Figure 2 (F_{att}). The resultant force is a vector resulting from the combined action of the attractive and repulsive fields, whose direction is the moving direction of the robot. The force analysis diagram of the artificial potential field is shown in Figure 2.

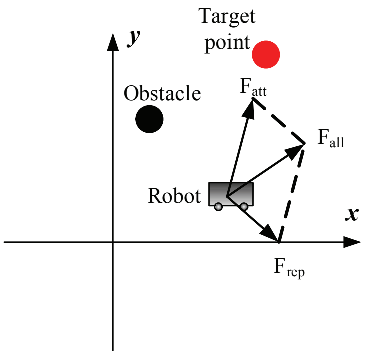


Figure 2: Force analysis diagram of classic artificial potential field

2.1.1 Gravitational Field

The gravitational potential field represents the attractive force of the target point on the robot. The further the robot is from the target point, the stronger the attraction, and vice versa. When the robot reaches the target point, the distance is zero, and the attraction is also zero. Therefore, the gravitational potential field function can be represented as:

$$\mathrm{U}_{\mathrm{att}}(q)=\frac{1}{2}\gamma\rho^{2}(q,q_{g}) \tag{9}$$

Where ρ represents Euclidean distance, γ is the proportionality coefficient, q is the current position coordinates of the robot, and q_g is the target point position. Based on the gravitational potential field function, the corresponding gravitational force is established. The gravitational force is the negative gradient of the gravitational potential field, which can be expressed as:

$$F_{att}(q)=-\Delta U_{att}(x)=\gamma\rho(q,q_{g}) \tag{10}$$

2.1.2 Repulsive Field

The repulsive potential field represents the repelling force exerted by the target point on the robot. The further the robot is from the target point, the smaller the repelling force it experiences, and vice versa; the closer the distance, the greater the repelling force[14]. When the robot moves beyond the influence range of the repulsive field around an obstacle, the repelling force from the obstacle on the robot becomes zero. Therefore, the function of the repulsive potential field can be expressed as:

$$U_{req}(x)=\begin{cases}\frac{1}{2}k(\frac{1}{\rho(q,q_{0})}-\frac{1}{\rho_{0}})^{2},0<\rho(q,q_{0})\le\rho_{0}\\ 0 \qquad ,\rho(q,q_{0})\ge\rho_{0}\end{cases} \tag{11}$$

Here, k is a positive proportionality constant, which is a constant that represents the maximum distance of influence of the obstacle's repulsive field.

According to the negative gradient of the repulsive potential field function, the repulsive force can be expressed as:

$$F_{req}(x)=\begin{cases}k(\frac{1}{\rho(q,q_{0})}-\frac{1}{\rho_{0}})\frac{1}{\rho^{2}(q,q_{0})}\Delta\rho(q,q_{0}),0<\rho(q,q_{0})\le\rho_{0}\\ 0 \qquad ,\rho(q,q_{0})\ge\rho_{0}\end{cases} \tag{12}$$

2.1.3 Resultant Force Field

During the movement of the robot, it is influenced by potential fields from multiple obstacles and target points. The resultant force on the robot is the combined effect of several repulsive and attractive forces[15], that is, the potential field experienced by the robot is:

$$U_{all}(p)=U_{att}(p)+U_{rep}(P) \tag{13}$$

According to the equation above, the total force acting on the robot is:

$$F_{all}(p)=F_{att}(p)+F_{rep}(P) \tag{14}$$

2.1.4 Limitations of the Classical Artificial Potential Field

The main shortcomings of the classic artificial potential field method are two-fold: unreachable goals and local minimum issues. When robots encounter these problems during path planning, it can lead to failures in path planning, preventing them from successfully completing mission objectives.

(1) Target Unreachable Issue

From the formulas of the repulsive and attractive fields, it is known that as the robot gets closer to the target point, the attraction exerted by the target point on the robot decreases, while the repulsion increases as the distance to obstacles decreases. When the robot, the target point, and the obstacle are aligned on a straight line, and as the robot's distance to both the target point and the obstacle decreases, the attraction diminishes and the repulsion increases. Near the target point, the total force acting on the robot becomes zero, preventing the robot from reaching the target point. This issue is referred to as the target unreachable problem.

(2) Local Minimum Problem

During the robot's navigation, it is influenced by the repulsive force from one or multiple obstacles and the attractive force from the target point. If the magnitude of the repulsive force equals the magnitude of the attractive force and their directions are opposite, the resultant force acting on the robot becomes zero. In this situation, the robot gets trapped in a local minimum of the potential field and is unable to escape from this local minimum point, leading to a failure in path planning.

2.2 Improved Artificial Potential Field Method

2.2.1 Improvement of the Potential Field Function

Due to the issues of local minima and target unreachability in the classical artificial potential field method, the following improvements are made: incorporating the repulsive force function of obstacles into the direction of attraction to guide the robot's movement.

$$F_{req1} = k(\frac{1}{\rho(q,q_0)} - \frac{1}{\rho_0})\frac{\rho_g^n}{\rho^2(q,q_0)} \tag{15}$$

$$F_{req2} = \frac{1}{2}nk(\frac{1}{\rho(q,q_0)} - \frac{1}{\rho_0})^2 \rho_g^{n-1} \tag{16}$$

Considering the impact of the robot's velocity on its movement during operation, the influence of velocity on the repulsive force function is increased, as shown in Equation 17.

$$F_v = \begin{cases} \mu v e^{(\frac{1}{\rho(q,q_0)} - \frac{1}{\rho_0})} & ,0 < \rho(q,q_0) \le \rho_0 \\ 0 & ,\rho(q,q_0) \ge \rho_0 \end{cases} \tag{17}$$

Among them, μ is a proportional factor, and v is the velocity of the robot.

2.2.2 Adaptive Step Length

The setting of the step length for robot movement is an especially important parameter in robot path planning. A step length that is too short results in inefficient path planning, while a step length that is too long can lead to collisions with obstacles or oscillations at the target point, failing to meet the requirements of robot path planning[16]. An adaptive variable step length is set so that when the robot is close to an obstacle, a smaller step length is used to escape the influence range of the obstacle. When the robot is far from the obstacle, a larger step length is adopted to improve the efficiency of robot path planning. When the robot approaches the target point, a smaller step length is used to reach the target point more precisely. The improved step length function is shown in Equation 18:

$$l = \begin{cases} \alpha l_0 dv & ,d < d_{\min} \\ \beta l_0 dv & ,d < \rho \\ l_0 & ,d > \rho \end{cases} \tag{18}$$

Among them,

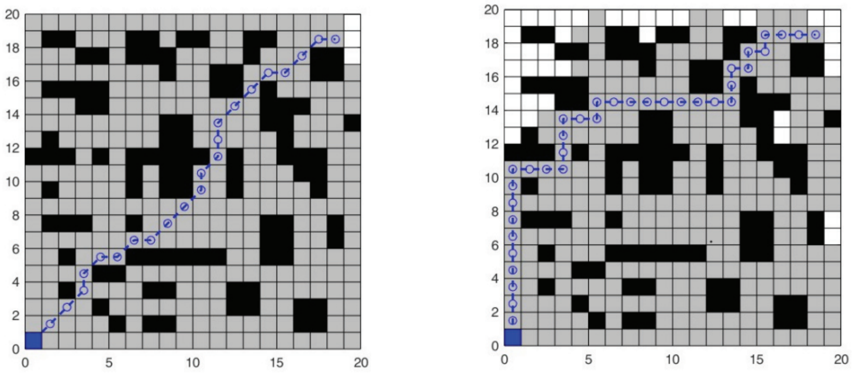
l_0 is the basic step length for escaping, α and β are proportional factors, v is the velocity of the robot's movement, $d_{\min}$ is the set minimum distance between the robot and the obstacle, and

ρ is the range within which the robot enters the repulsive influence of the obstacle.

3. Algorithm Simulation

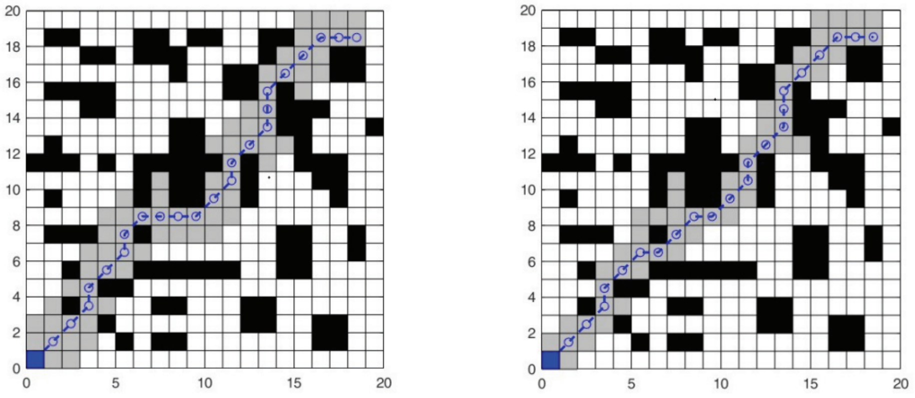
3.1 A* Algorithm Simulation

To verify the effectiveness of the improved A* algorithm, a simulation is implemented using the MATLAB 2023b compilation environment. The simulation is shown in Figure 3, where black represents obstacles, gray represents search nodes, the blue line represents the planned path, and the small circles on the blue line are key nodes searched by the algorithm. Figure 3-a shows the simulation of the Dijkstra algorithm, Figure 3-b shows the simulation of the A* four-neighborhood search algorithm, Figure 3-c shows the simulation of the A* eight-neighborhood search algorithm, and Figure 3-d shows the simulation of the improved A* search algorithm presented in this paper. A comparison of the simulations of each algorithm is shown in Table 1.



(a)

(b)



(b)

(d)

Figure 3: Algorithm Simulation Diagram

Table 1: Algorithm Simulation Comparison

Algorithm	Path	Key Points	Simulation Time	Turning Angle
Dijkstra Algorithm (a)	27.799	310	7.004	585
A* 4 neighborhood algorithm (b)	36	274	3.860	990
A* 8 neighborhood algorithm (c)	28.385	70	1.279	495
improves the A* algorithm	27.799	53	1.164	495

Figure 3 and Table 1 show that the simulation results indicate that, compared to the Dijkstra algorithm, the algorithm presented in this paper is capable of finding the optimal path while reducing the number of search nodes, simulation time, and turning angles by 82.9%, 83.42%, and 15.38% respectively. Compared to the A* four-neighborhood algorithm, there is a reduction in the optimal path distance, number of search nodes, simulation time, and turning angles by 22.78%, 80.65%, 69.84%, and 50% respectively. In contrast to the A* eight-neighborhood algorithm, there is a decrease in the optimal path distance, number of search nodes, and simulation time by 2.06%, 24.26%, and 8.99% respectively. Data analysis indicates that the improved algorithm optimizes various aspects including the optimal path, number of search nodes, simulation time, and turning angles when compared to some algorithms. It can be observed that the improved A* algorithm still contains redundant key nodes; hence, further optimization is conducted for secondary path planning to eliminate superfluous key nodes and smooth the path. The simulation is shown in Figure 4 and Table 2.

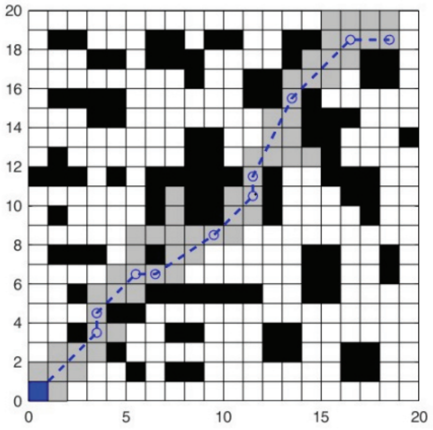


Figure 4: Secondary Optimization Simulation

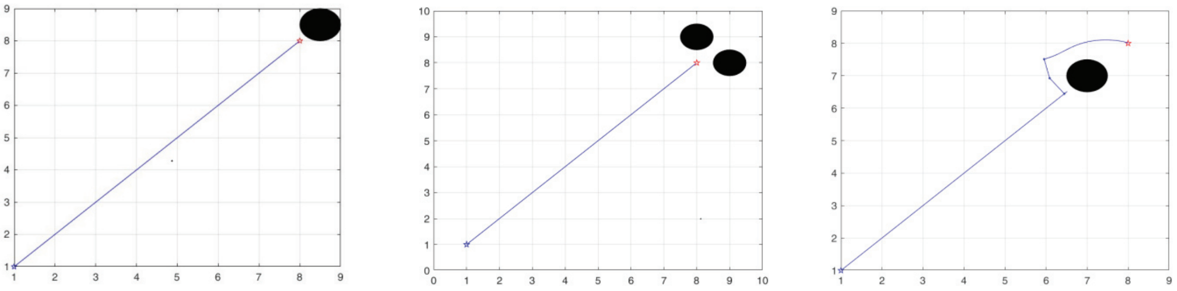
Table 2: Data for Secondary Optimization Simulation

Algorithm	Path	Key Points	Simulation Time	Turning Angle
Improving the algorithm	27.219	10	1.058	315

The simulation results demonstrate that the secondary improved algorithm reduces the optimal path, simulation time, and turning angles by 2.08%, 9.1%, and 36.36% respectively compared to the original improved algorithm. The number of path keypoints decreases from 22 to 10, a reduction of 54.54%. After smoothing, a more optimal path can be planned, with significantly reduced turning angles, making it more suitable for practical robotic applications, reducing the difficulty of robot control, and enhancing safety.

3.2 Artificial Potential Field Method Simulation

To address the issues of local minima entrapment and unreachable targets commonly associated with the artificial potential field method, this paper modifies the potential field function and incorporates an adaptive step size to escape local minima and reach unattainable targets. The simulation is shown in Figure 5, where Figure 5-a illustrates the correction for unreachable target issues, while Figures 5-b and 5-c demonstrate escaping local minima problems.



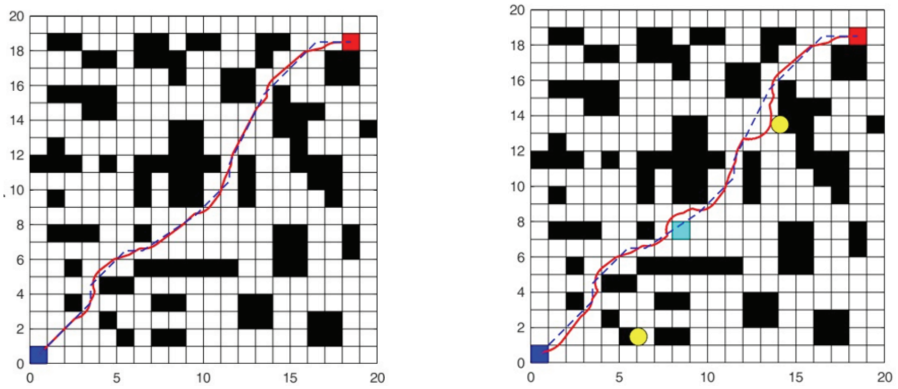
(a) (b) (c)

Figure 5: Improved Artificial Potential Field Method Simulation

3.3 Fusion Algorithm Simulation

Limited by the A* algorithm, the improved A* algorithm can only plan an optimal path in a known static map. When obstacles are added to a known map or in dynamic maps, the A* algorithm cannot meet the planning requirements. The artificial potential field method is a local path planning algorithm that plans a locally optimal path, but it cannot find the globally optimal path. Moreover, when escaping from special environments, the artificial potential field method may encounter issues such as long path planning times and low planning efficiency. To address these issues, this paper combines the improved A* algorithm with the improved artificial potential field method for robot path planning. The fused algorithm not only ensures global optimality of the path but also achieves local obstacle avoidance in unknown and dynamic environments. By introducing the A* algorithm into the artificial potential field method, using the key nodes of the A* algorithm to provide directional guidance for escape in the artificial potential field method, it effectively escapes local minima and unreachable targets, resulting in higher planning efficiency.

As shown in Figure 6, where blue represents the starting point, red represents the endpoint, green represents static obstacles, yellow represents dynamic obstacles, and the red line represents the path planned by the fusion algorithm. Figure 6-a shows the path planning by the fusion algorithm, Figure 6-b shows the path planning with both static and dynamic obstacles added.



(a) (b)

Figure 6: Fusion Algorithm Simulation

The simulation demonstrates that the fused algorithm is capable of finding a planned path from the starting point to the endpoint, addressing the limitations of the A* algorithm in avoiding static and dynamic obstacles, and smoothing the globally optimal path of the A* algorithm. By incorporating local path planning using the artificial potential field method, the robot can maximally avoid obstacles and effectively prevent collisions with them.

4. Conclusion

This paper proposes a robot path planning approach based on an improved A* algorithm and artificial potential field method. The A* algorithm is enhanced through search neighborhood, heuristic function, and redundant node optimizations. The results indicate that the improved A* algorithm significantly improves the number of search nodes, simulation time, and turning angles while ensuring an optimal path, making it more suitable for global path planning of robots in static environments. The artificial potential field method is refined by modifying the potential field function, introducing adaptive step sizes, and integrating key nodes from the A* algorithm to overcome issues of local minima and unreachable targets, achieving local path planning. The fusion algorithm enables robot motion planning in unknown environments, satisfying both global optimal path planning and local robot motion obstacle avoidance, maximally avoiding obstacles, and enhancing the safety of robot motion. Future work will incorporate more intelligent path planning algorithms to further improve the level of robot intelligence.

Acknowledgements

Research project "Research on Industrial Robot Fault Diagnosis Technology Based on Industrial Internet", project No. 2023AH052918, supported by "Research Center for New Energy Vehicle Motor Drive Technology of Anhui Institute of Information Engineering's School level Science and Technology Innovation Platform", "Research Team for Automotive Electronic Control System of Anhui Institute of Information Engineering's School level Scientific Research Team.

References

[1] Miyombo Ernest Miyombo, Yong-kuo Liu, Chishinga Milton Mulenga, et al.Optimal path planning in a real-world radioactive environment: A comparative study of A-star and Dijkstra algorithms[J].Nuclear Engineering and Design, 2024, 420(1): 113039.

[2] Chi Xu, Li Hua, Fei Jiyou.Research on Robot Random Obstacle Avoidance Method Based on the Fusion of Improved A * Algorithm and Dynamic Window Method [J].Journal of Instrumentation, 2021, 42(3):132-140.

[3] Kai Dong, Dewei Yang, Jinbao Sheng, et al. Dynamic planning method of evacuation route in dam-break flood scenario based on the ACO-GA hybrid algorithm[J].International Journal of Disaster Risk Reduction, 2024, 100(1):104219.

[4] Mohd Nadhir Ab Wahab, Amril Nazir, Ashraf Khalil, et al.Improved genetic algorithm for mobile robot path planning in static environments[J].2024, 249(1): 123762.

[5] Bodong Tao, Jae-Hoon Kim. Mobile robot path planning based on bi-population particle swarm optimization with random perturbation strategy[J].Journal of King Saud University - Computer and Information Sciences, 2024, 36(2):101974.

[6] R. Szczepanski.Safe Artificial Potential Field - Novel Local Path Planning Algorithm Maintaining Safe Distance From Obstacles[J].2023, 8(8):4823-4830.

[7] Yin X, Cai P, Zhao K, Zhang Y, et al. Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusionalgorithms[J].Sensors, 2023, 23(8):4102.

[8] II-kyu Ha.Improved A-Star Search Algorithm for Probabilistic Air Pollution Detection Using UAVs[J].sensors, 2024, 24(4):1141.

[9] Raihan Kabir, Yutaka Watanobe, Md Rashedul Islam, et al. Enhanced Robot Motion Block of A-Star Algorithm for Robotic Path Planning[J].Sensors 2024, 24(5):1422.

[10] Eugene Auh, Juhwak Kim, Younghwan Joo, et al.Unloading sequence planning for autonomous robotic container-unloading system using A-star search algorithm[J]. Engineering Science and Technology, an International Journal, 2024, 50(1):101610.

[11] Tiangen Chang, Guofu Tian.Hybrid A-Star Path Planning Method Based on Hierarchical Clustering and Trichotomy[J].Appl. Sci. 2024, 14(13):5582.

[12] Antonios Chatzisavvas, Michael Dossis, Minas Dasygenis.Optimizing Mobile Robot Navigation Based on A-Star Algorithm for Obstacle Avoidance in Smart Agriculture[J].Electronics 2024, 13(11):2057.

[13] Ju Gao; Xiangrong Xu; Quancheng Pu, et al.A Hybrid Path Planning Method Based on Improved A* and CSA-APF Algorithms[J]. IEEE Access, 2024, 12(1):39139-39151.

[14] Cao, R, Guo, Y, Zhang, Zet al. Global path conflict detection algorithm of multiple agricultural machinery cooperation based on topographic map and time window[J].Comput. Electron. Agric. 2023, 208:107773.

[15] Yuheng Wei, Xinguo Wei, Hao Liu.A Star Identification Graph Algorithm Based on Angular Distance Matching Score Transfer[J].2024, 24(5):6539-6547.

[16] Tao WANG, Gangyi WANG, Xinguo WEI, et al.A star identification algorithm for rolling shutter exposure based on Hough transform[J]. Chinese Journal of Aeronautics, 2024, 37(6):319-330.