Journal of Physics: Conference
Series

Mobile Robot Path Planning Algorithm Based on Improved A Star

To cite this article: Lei Zhang and Yuan Li 2021 *J. Phys.: Conf. Ser.* **1848** 012013

Mobile Robot Path Planning Algorithm Based on Improved A Star

Lei Zhang[1] **, Yuan Li**[1*]

[1]Electronic Information School, Shanghai Dianji University, Shanghai, 201306, China

[*]Corresponding author's e-mail: yli@sdju.edu.cn

Abstract. With the rapid development of artificial intelligence, path planning has become one of the key technologies in the application of mobile robots. Although the A star(A*)path planning algorithm can plan a collision-free path, there are still shortcomings such as the path is not optimal, too many redundant points, and not smooth enough. Aiming at the deficiencies of the A* algorithm, an improved A* algorithm is proposed. Firstly improved A* algorithm divides the distance between adjacent nodes in the path planned by A* algorithm. Then we use the two-way optimization method to delete useless nodes, short the path length and reduce the number of inflection points. Finally, a cubic spline function is introduced to smooth the path. The simulation results show that the improved A* algorithm is effectively optimized in terms of path length, number of inflection points and smoothness.

1. Introduction

With the popularity of mobile robots in daily life, how to find a fast and efficient path has become a hot research direction. If a mobile robot wants to quickly complete the assigned task, it needs to plan an optimal path. Path planning refers to obtaining environmental information through sensors in a complex work environment, and using this information to find a collision-free safe path from the start point to the end point for the mobile robot [1]. Path planning is divided into global path planning and local path planning according to the known degree of the environment. Commonly used path planning methods include Dijkstra algorithm [2], artificial potential field method [3] and A* algorithm [4]. The A* algorithm has attracted wide attention because of its simple principle, easy implementation, high efficiency, fast speed and the ability to search for the optimal solution. It is considered to be the most effective direct search algorithm for solving the shortest path in a static road network. [5-6]. On the basis of the Dijkstra algorithm, Nilsson proposed the A* algorithm in 1980, which is a global path planning based on heuristic search. The disadvantage is that there are many turning points and non-optimal paths [7]. Peng Song et al. used a secondary search strategy in the literature [8] to optimize the planned path, but did not consider that the number of path inflection points is still large. Chen Ruonan et al. combined the angle information and the partition adaptive distance information in the literature [9] to improve the heuristic function to improve the calculation efficiency, but the path is still not optimal. The above literature shows that the A* algorithm has been improved by many scholars, but the following problems have not been well solved: the planned path is not optimal and there are many inflection points. Aiming at the above problems, this paper designs an improved A* algorithm, which uses the division of adjacent nodes and two-way optimization methods to reduce the path planning length and the number of inflection points.

2. Improved A* path planning algorithm design

2.1. Traditional A algorithm*

The A* algorithm is a heuristic search algorithm that comprehensively evaluates the cost value of each extended search node by setting a suitable heuristic function. By comparing the cost value of each expansion node, choose the most promising point to expand until the target node is found [10]. Since the node with the lowest estimated value is used as the expansion node each time, the final path cost is the lowest [11]. The valuation function is as follows:

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

Where, f(n) represents the evaluation function of node n; g(n) represents the actual cost from the starting node to the n node; h(n) is the estimated cost from the n node to the target node. When h(n)=0 in the evaluation function, formula (1) becomes Dijkstra's algorithm, which can find the shortest path, but needs to calculate a large number of nodes. When g(n)=0, the A* algorithm uses the best priority search of the greedy strategy, and the calculation speed is fast but the shortest path is not necessarily found. This article uses Euclidean distance to estimate the cost:

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

Where, (x_g,y_g) is the current node position, (x_n,y_n) is the target node position.

The A* algorithm uses the evaluation function to find the shortest path and requires continuous updating of two lists: the close list and the open list. In the search process, all nodes around the current node are stored in the open list. The A* algorithm will select the smallest estimated value as the next expansion node and put the node in the close list. Repeat the above process until the target node is added to the open list and the final path is obtained, then the path search ends.

2.2. Improved A algorithm*

Although the path planned by the A* algorithm can find a safe path, there are many inflection points, the path is not optimal, and the path is not smooth enough. Applying the A* algorithm to a mobile robot that is used in an actual environment will waste a lot of time and reduce the speed of exercise. This paper proposes an improved A* algorithm to solve the above problems.

In order to observe more intuitively, this article uses raster map to study path planning. Figure 1 is a local environment map of a mobile robot, in which black grids represent obstacles (obstacles are expanded and cars are treated as mass points), white grids represent passable areas and black lines represent paths. The improved A* algorithm first divides the distance between every two adjacent nodes on the path planned by the A* algorithm. Then the nodes are removed according to a certain rule, and the reserved nodes are sequentially connected to obtain a planned path. On the basis of the obtained path, this operation is performed from the target point to the starting point for secondary path planning, and the optimal path of the improved A* algorithm is obtained. Specific steps are as follows:

Step 1: Select the starting point and the end point and use the A* algorithm to plan a partial path map in any environment (as shown in figure 1). The path planned by the traditional A* algorithm has n nodes, and the distance between every two adjacent nodes is divided. The distance between adjacent nodes is equally divided into k (k=2) equal parts, and the path contains ((n-1)*k+1) nodes.

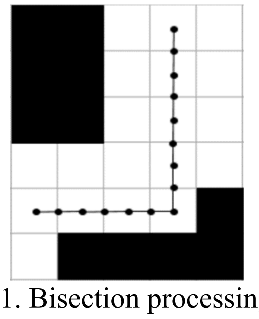

Figure1. Bisection processing path

Step 2: B is the set of newly generated path nodes, first store the starting node in set B, starting node s=0, target node d=(n-1)*k. Connect the starting node with the target node. If you cross the obstacle, perform the d-1 operation to make the node continue to be connected with the current node to judge whether to cross the obstacle. The node is stored in set B until the obstacle is not crossed, and the node is set as the starting node. Continue to repeat the above operations until the target node is stored in set B, and the elements in set B are connected in sequence. The one-way planning path is shown as the dotted line in figure 2.

Step 3: Perform k (k=2) equal division processing on the distance between adjacent nodes for the path obtained in step 2, and then optimize from the target point to the starting point (such as step 2). Improve the A* algorithm to finally optimize the path (shown in the dotted line in figure 3).

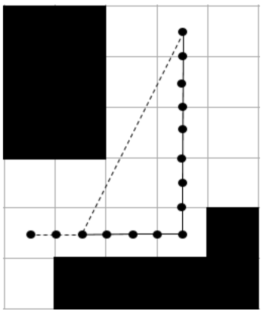

Figure2. One-way planning path

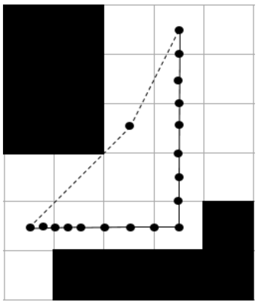

Figure3. Path to improve A* planning

The solid line in figure 3 is the path planned by the A* algorithm, and the dotted line is the path planned by the improved A* algorithm. It can be intuitively observed that the improved A* algorithm is better than the A* planned path in terms of path length.

The improved A* algorithm adopts the idea of dividing the distance between adjacent nodes and two-way optimization of the path planned by the A* algorithm. The obstacles are expanded and the mobile robot is regarded as a mass point. Make the path node no longer limited to the grid center, reduce the number of inflection points and path length, and make the path smoother.

3. Path smoothing

The improved A* path planning algorithm optimizes the problems of a large number of inflection points and a long path. However, because the A* algorithm is simulated in a grid map, the problem of inflection point spikes still exists. This paper uses cubic B-spline curve to smooth the path [12].

Assuming that given k+1 control points, the k-th B-spline curve equation is:

$$p(t)=\sum_{i=1}^{n} P_i F_{i,k}(t) \qquad t\in[0,1] \tag{3}$$

Where, P_i is the characteristic point of the control curve, $F_{i,k}(t)$ is the basis function of the B-spline curve, and the polygon formed by its vertices P_i (i=0,1,…,n) is called the B-spline The characteristic polygon of the curve.

$$F_{i,k}(t)=\frac{1}{k!}\sum_{j=1}^{k-i}(-1)^j C_{k+1}^{j}(t+k-i-j)^k \tag{4}$$

Where,0<t<1,i=0,1,…,k-1.

$$C_{k=1}^{j}=\frac{(n+1)!}{\left(n+1-j\right)!} \tag{5}$$

When k=3, the cubic B-spline curve equation is:

$$P(t)=\sum_{i=1}^{n} P_i F_{i,3}(t) \tag{6}$$

The basis function expression is:

$$\begin{cases} F_{0,3}(t)=\frac{1}{6}(-t^3+3t^2-3t+1) \\ F_{1,3}(t)=\frac{1}{6}(3t^2-6t+1) \\ F_{2,3}(t)=\frac{1}{6}(-3t^3+6t^2+3t+1) \\ F_{3,3}(t)=\frac{1}{6}t^3 \end{cases} \tag{7}$$

Bring formula (7) back to formula (3) to get:

$$P_{0,3}(t)=\frac{1}{6}\begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{8}$$

Combining the cubic B-spline curve with the global path A* algorithm, and calculating by the above formula, the smoothed path $P_0,\ldots,P_n$ can be obtained. The smoothed path is helpful to reduce the stuck phenomenon of the mobile robot in the driving process.

4. Algorithm simulation and analysis

This experiment uses the Intel Core i5-4200M processor, the main frequency is 2.5GHz, and the memory is 8GB. A 40×40 grid map was established in the Python environment, and simulation experiments were carried out. It is verified that the improved A* algorithm has certain advantages over the A* algorithm in the number of inflection points and the path length.

4.1. Comparison and analysis of improved A algorithm and traditional A* algorithm*

In the simulation environment, the black grid is the obstacle area, and the white grid is the free passable area. the polyline is the planned path. Although the A* algorithm can effectively avoid obstacles, the path is obviously not optimal, as shown in figure 4. The improved A* path planning algorithm has been significantly optimized by dividing the distance between adjacent nodes and bidirectional optimization, as shown in figure 5.

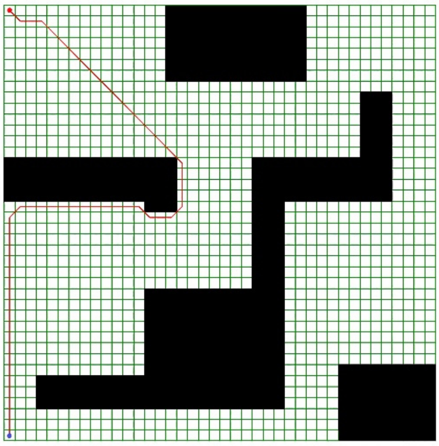

Figure4. Traditional A* algorithm path

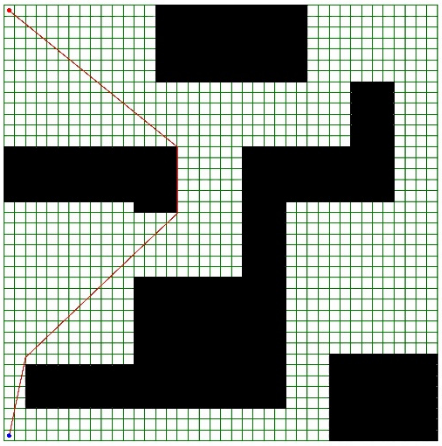

Figure5. Improved A* algorithm path

It can be seen from the comparison of simulation experiments in figure 4 and figure 5. Compared with the A* algorithm, the improved A* path planning algorithm proposed in this paper is no longer limited to the grid center, and the path length and the number of inflection points are significantly reduced.

Table 1 shows the performance index comparison of the two paths running from the starting point to the target point in a grid environment. It can be seen from Table 1 that the improved A* path planning algorithm effectively shortens the path length and reduces the number of inflection points.

Table 1. Comparison of various values of A* algorithm and improved A* algorithm.

Performance	A* algorithm.	Improved A* algorithm.
Path length	63.038	53.616
Number of inflection points	9	3
Relative A* algorithm path length reduction rate/%	-	15.8
Relative A* algorithm inflection point reduction rate/%	-	66.7

Under the condition that the grid environment, starting point and target point are the same. It can be seen from Table 1 that the number of inflection points and path length of the improved A* path planning algorithm are reduced by 66.7% and 15.8% respectively compared with the A* algorithm.

4.2. Path smoothing simulation

In the grid environment, the improved A* path planning algorithm is used to plan a path for the mobile robot, and then the planned path is smoothed with the cubic B-spline method. The simulation result is shown in figure 6.

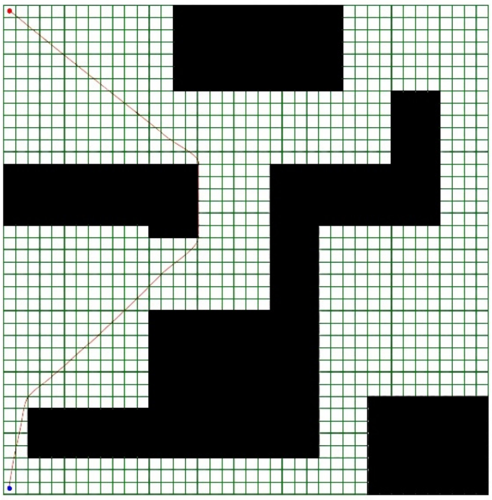

Figure6. Smoothing path

Although there is still a spike problem at the inflection point of the improved A* path planning algorithm, after smoothing the inflection point by introducing the cubic B-spline, it can be better applied in the actual environment. The smoothed path is connected by a number of smooth and continuous curves, so that the mobile robot maintains a certain speed at corners and runs more smoothly.

5. Conclusion

This paper uses the improved A* algorithm to plan the path of the mobile robot, and optimizes the A* algorithm to plan the path, which has many inflection points and longer paths. Obstacle expansion guarantees a safe distance between the planned path and the obstacle. Two-way optimization processing makes the path nodes not necessarily limited to the grid center, while reducing the path length and the number of inflection points. The smooth processing of the inflection point is helpful for the mobile robot to optimize the turning speed and the stall at the inflection point in the actual environment.

References

[1] Xu, X., Fan, X.W., Xu, L. (2019) Mobile robot path planning based on BCA* algorithm. J. Computer Simulation., 36(10):322-328.

[2] Yuan, B., Liu, J.S., Qian, D., Luo, D.H. (2014) Analysis of a logistics network path optimization algorithm based on improved Dijkstra. J. Manufacturing Automation., 36(09): 8688+105.

[3] Yu, Z.Z., Yan, J.H., Zhao, J., Chen, Z.F., Zhu, Y.H. (2011) Improved artificial potential field method for mobile robots Path planning. J. Journal of Harbin Institute of Technology., 43(01):5055.

[4] Zhao, W.D., Jiang, C. (2020) Two-stage search A* global path planning algorithm. J. Computer Application and Software Pieces., 37(12): 249253.

[5] Qiao, Y.X., Wang, Q., Yang, Y., Zhang, Y. (2020) Smooth path planning based on A* algorithm for searching for back obstacles. J. Sensors and Microsystems., 39(08):127129.

[6] Zhang, H.M., Li, M.L., Yang, L. (2018) Mobile robot safe path planning based on improved A * algorithm. J. Computer Simulation., 35(04):319324.

[7] Wang, W., Pei, D., Feng, Z. (2018) Improved A* algorithm for shortest path planning of mobile robots. J. Journal of Computer Applications., 38(5):15231526.

[8] Peng, S., Jia, Y. (2010) Improvement of A* Algorithm in Global Path Planning of Lunar Rover.

J. Spacecraft Engineering., 19(4):8085.

[9] Chen, R.N., Wen, C.C., Peng, L. (2019) Application of improved A* algorithm in robot indoor path planning. J. Journal of Computer Applications., 39(4):10061011.

[10] Zhang, G.L., Hu, X.M., Chai, J.F., Zhao, L., Yu, T. (2011) Overview of path planning algorithms and their applications. J. Modern Machinery., (05): 8590.

[11] Sun, W., Lu, Y.F., Tang, H.W., Xue, M. (2017) Mobile robot path planning based on an improved A* algorithm. J. Journal of Hunan University (Natural Science Edition)., 44(04):94101.

[12] Huang, C., Fei, J.Y., Liu, Y., Li, H., Liu, X.D. (2017) A smooth path planning method based on dynamic feedback A* ant colony algorithm. J. Transactions of the Chinese Society of Agricultural Machinery., 48(04): 3440+102.