Scholars Journal of Engineering and Technology
Abbreviated Key Title: Sch J Eng Tech
ISSN 2347-9523 (Print) | ISSN 2321-435X (Online)
Journal homepage: https://saspublishers.com

OPEN ACCESS

Research on AGV Path Planning based on Improved A* algorithm

Wang Sen Lin[1*]

[1]School of Information Engineering, Shenyang University of Chemical Technology, Tiexi, Shenyang, China, 110142

DOI: 10.36347/sjet.2023.v11i03.004 | **Received:** 15.02.2023 | **Accepted:** 22.03.2023 | **Published:** 24.03.2023

***Corresponding author:** Wang Sen Lin
School of Information Engineering, Shenyang University of Chemical Technology, Tiexi, Shenyang, China, 110142

Abstract **Review Article**

Path planning is an important field for AGV, and A* algorithm is one of the fastest shortest path algorithms, which is often used in a variety of path planning methods. This paper proposes an improved A* algorithm based on path planning in grid map, aiming at the low efficiency of traditional A* algorithm in searching the map and the existence of many redundant points and turning points in the path. The algorithm uses the weighted Manhattan distance as the heuristic function, and the coefficient changes with the distance. In addition, a turning penalty mechanism is added to the heuristic function to reduce the occurrence of turning points. Finally, a reduction strategy is added to the search function based on the location of the current point and the end point to reduce the search of redundant points. The results of python simulation experiments show that this improved A* algorithm can effectively reduce the redundant points and turns in the search, and improve the search efficiency and the smoothness of the path in the path planning.

Keywords: AGV, path planning, A* algorithm, heuristic function.

INTRODUCTION

AGV (Automated Guided Vehicle) refers to a robot transport vehicle with an automatic navigation device that can travel along a specified navigation path. It is used as an important logistics transportation carrier in intelligent manufacturing workshop because of its simple structure and convenient operation [1]. Facing the complex and changeable outdoor environment, improving the effectiveness and smoothness of vehicle path planning through algorithms has become one of the main research contents of path planning.

Global path planning is to plan the path in a known environment. The environmental information has been given and the information such as the location of the obstacle does not change. Generally, it only needs to work according to the Plan in advance global path [2]. The global algorithm includes Dijkstra algorithm [3], genetic algorithm [4], A* algorithm and so on. The Dijkstra algorithm draws on the idea of greedy algorithm, and searches outward layer by layer with the starting point as the center point, until the target point is searched or all nodes are traversed, and the optimal path from the starting point to all points (including the end point) is obtained [5]. Genetic algorithm is one of the intelligent algorithms. Because genetic algorithm has strong adaptability to different spaces, it is widely used to solve path planning problems. However, there are also problems such as low initial population quality and slow convergence speed [6]. The A* algorithm is the most mature and widely used algorithm now. By referring to the heuristic function, the relationship between the running speed and the optimal solution is balanced, but there are also problems such as low computational efficiency, redundant points and inflection points, and the path is not smooth [7]. In response to the above problems, researchers have improved the A* algorithm from multiple perspectives. Aiming at the excessive turning times of A* algorithm, Wang Hongwei [8] proposed to delete the intermediate nodes when there are no obstacles on the connection between the front and back nodes, thus optimizing the path smoothness, but the computational efficiency is not improved. Aiming at the problem of path search efficiency of A* algorithm, Gao Qingji [9] introduced 'artificial search flag'to reduce the search of useless intervals and improve the search efficiency. Wu Peng [10] used a bidirectional A* search strategy to search at the starting point and the end point respectively. The two paths converge in the middle area to obtain a complete search path, which improves the search efficiency of the path, but the obtained path is not necessarily the optimal path. In Reference [11], the selection rules of child nodes are added to the safety of mobile robots for planning paths, so as to avoid mobile robots obliquely crossing obstacle vertices and reduce collision risks, but the optimal path cannot be obtained by increasing the length of the path.

In order to solve the shortcomings of traditional A* algorithm in AGV path planning, such as slow search speed and many turning points, an improved A* algorithm is proposed. By weighting the heuristic function and simplifying the search points, the speed is improved, and the turning penalty mechanism is introduced to reduce the number of turns. By using python resume grid map for simulation, the search time and the number of turns can be reduced.

1. Traditional A* algorithm

A* algorithm is improved by Dijkstra algorithm. It is a heuristic algorithm for optimal search in static environment. It has good search efficiency and accuracy and is often used in path planning. Compared with the Dijkstra algorithm's aimless search, it can indicate a direction from the starting point to the end point through its own cost function, so it is more efficient than the Dijkstra algorithm. It evaluates the consumption of the current point through a heuristic function for path planning. Its heuristic function is:

$$f(n)=g(n)+h(n) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

In the formula, g(n) represents the actual cost of the starting point moving to the specified location, h(n) is the estimated cost of the current point moving to the end point, which is the heuristic function of the algorithm, and f(n) is the total cost of the point.

The selection of the heuristic function is also crucial, and the behavior of the A* algorithm can be adjusted by modifying the heuristic function. Assuming that the real path cost of the robot from the current point to the target point is H, there are the following situations:

1) When H > h(n), we can ensure to find the optimal path, but the smaller the estimated value of H is, the more nodes it searches and traverses, and the lower the overall efficiency of the algorithm is.
2) When H = h(n), the algorithm can calculate the optimal path, and will not traverse the redundant nodes. However, due to the existence of obstacles in practice and the uncertainty of obstacles, it is difficult to achieve the ideal situation.
3) When H < h(n), the algorithm can not guarantee to calculate the optimal path, but the efficiency will be the highest. This scheme can be selected when considering the non-optimal path and time efficiency.

The current heuristic function generally uses one of the Manhattan distance in Equation (2) and the Euler distance in Equation (3) in practical applications:

$$h(n)=|x_i - x_n| + |y_i - y_n| \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(2)}$$

$$h(n)= \sqrt[2]{(x_i - x_n)^2 + (y_i - y_n)^2} \quad \ldots\ldots\ldots\ldots \text{(3)}$$

The A* algorithm is based on the calculation of the evaluation function of the relevant nodes in the search path, so as to determine the direction of motion and the generation of the guide path. The starting point is put into the OPEN list as the first parent node, and the evaluation function of the surrounding 8 nodes is calculated. The node corresponding to the minimum evaluation function value is selected as the next parent node. At the same time, the starting point is put into the CLOSE list, and the new parent node is used to calculate the evaluation function value of the shield node. The OPEN and CLOSE lists are continuously updated to select the nodes corresponding to the minimum value of the evaluation function from the starting point, and the iteration is continued until the end point. Connect these selected nodes from the end point to the starting point in turn to form the entire optimal path. It is precisely because the minimum evaluation function value is selected in the search process, so the final path cost is the smallest.

2 Improvement of A* algorithm

2.1 Modification of heuristic function

When the Manhattan distance is used as the heuristic function h(n) of the A* algorithm, which is closer to the shortest distance from the current node to the target node, the algorithm search efficiency is higher. However, in practice, there is often a large difference between h(n) and the shortest distance from the current node to the target node.

In the process of AGV car running, if the current node is (x_n,y_n),the target node (x_{goal} ,y_{goal}),with (L_x , L_y)represents the actual length of the current point to the target node,in the process of A* algorithm running, there will be $|x_n$-$x_{goal}|$ less than L_x or $|y_n$-$y_{goal}|$ less than L_y.At this time, there will be h(n)< H,the algorithm traverses more nodes,which in turn affects the efficiency of the algorithm.

Aiming at the problem that the traditional A* algorithm has many traversal nodes in AGV path planning, this paper uses the weighted Manhattan distance as the heuristic function to improve the traditional A* algorithm. The improved heuristic function h1(n) is:

$$h_1(n) = (1+\frac{d(n)}{d(s)})h(n) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4)}$$

In the formula, the distance from the current node n to the end point is expressed, and the evaluation function of the distance is expressed by Euler distance. The distance from the starting point to the target node is still represented by the Euler distance. It can be seen from the formula that when the algorithm is initially running, the heuristic function h(n) is too large, and the algorithm will quickly move closer to the target node, and then gradually tend to find the shortest path.

The weighting of the heuristic function can improve the efficiency of the algorithm to a certain extent. However, the path planned by the improved A algorithm using the weighted Manhattan distance as the heuristic function still has the problem of more path turns. Based on the heuristic function, the turning correction cost is introduced. The specific operation is as follows:

Set the current node (x_n,y_n), its parent node (x_{n-1},y_{n-1}), child node (x_{n+1},y_{n+1}). Then judge whether the AGV turns and passes Formula(5) and Formula(6).

a_1=(x_n-x_{n-1})(y_{n+1}-y_n) ……………….. (5)

a_2=(x_{n+1}-x_n)(y_n-y_{n-1}) ………………. (6)

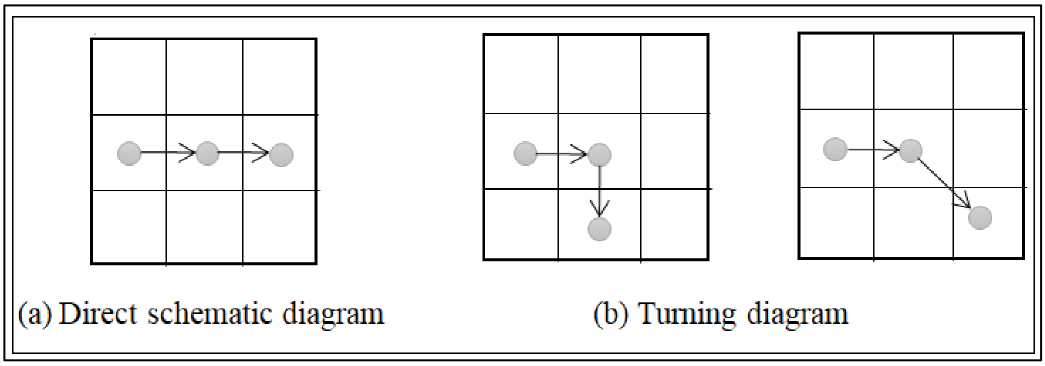

Fig 1: Node Location

In the figure are several cases of node location, from left to right once for the parent node, the current node, child nodes.

The heuristic function $h_1(n)$ obtained by integrating the above results is:

$$h_1(n)=\begin{cases}(1+\frac{d(n)}{d(s)})(|x_i-x_n|+|y_i-y_n|)+b & a_1\neq a_2\\ (1+\frac{d(n)}{d(s)})(|x_i-x_n|+|y_i-y_n|) & a_1=a_2\end{cases}$$

In the formula, b represents the turning correction coefficient, which ensures that the AGV is preferentially straight in the selected path.

H(n) introduces the turning correction cost parameter b, and the selection of b directly affects the size of h(n), thus affecting the size of the cost estimation function f(n). If the turning correction cost parameter is too large, AGV may have a detour phenomenon; if the turning correction cost parameter a is too small, it will not play a role in correction. Therefore, it is necessary to select the appropriate turning correction cost parameter b, which should not only ensure that AGV postgraduate entrance examination gives priority to straight line, but also avoid detour. The number of path turns calculated by the A^* algorithm is reduced.

2.2 Simplified search

The traditional A^* algorithm generally has eight search directions when expanding each node, which are front, back, left, right and four oblique angles. The center position of these nine nodes represents the current point, that is, the position of the car, and there are 8 nodes around it. It is necessary to calculate the evaluation function value of these 8 nodes, and select the node corresponding to the minimum value as the next node reached by the mobile robot, which leads to the traversal of many useless nodes in this process, resulting in low efficiency of the algorithm. At this time, using the simplification strategy [6], we can discard the expansion points that are not related to the path according to the positional relationship between the current point (x_n,y_n) and the end point (x_{goal},y_{goal}).If x = x_{goal}-x_n, y = y_{goal}-y_n, the rectangular coordinate system is established, and the coordinate system is divided into four regions and four axes. The value of x, y is used to determine the location area where the end point is located at the current node, and the points on the opposite area and the adjacent axis of the area are discarded. If the current node and the target point are on the same axis, that is, x= 0 or y = 0, the node is not deleted.

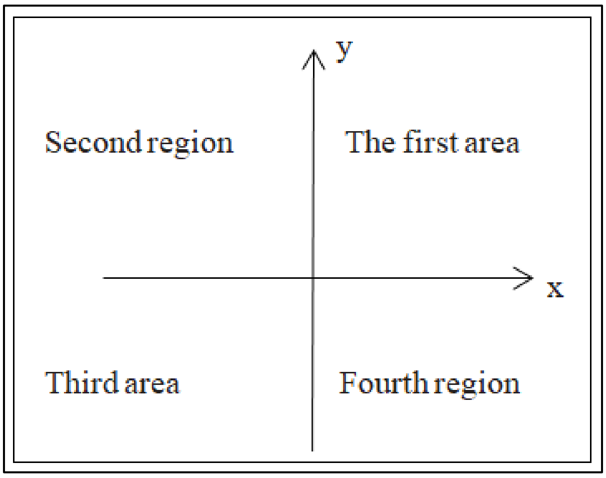

Fig 2: Search area graph

3. EXPERIMENTAL SIMULATION

3.1 Map modeling

The improved A^* algorithm in this paper uses python as the language and PyCharm as the software platform. The traditional A^* algorithm and the improved A^* algorithm are used to plan the path of the AGV car respectively, and the corresponding experimental results are obtained. The experimental results of the simulation are compared to verify the effectiveness of the A* algorithm.

In this paper, the environment map is constructed by 20*20 grid map, and the starting point and obstacle information are represented in the grid environment map.

3.2 Simulation experiment

In this paper, the traditional A* algorithm is simulated with the A* algorithm using heuristic function modification and streamlining search respectively. The A * algorithm grid diagram is shown in Figure 4. In Figure 4, the yellow grid represents the starting point, the green grid represents the end point, the white grid is the passable area, the black grid is the obstacle area, the gray-white grid is the grid that has been traversed using the A* algorithm, and the red represents the grid that has not been traversed.

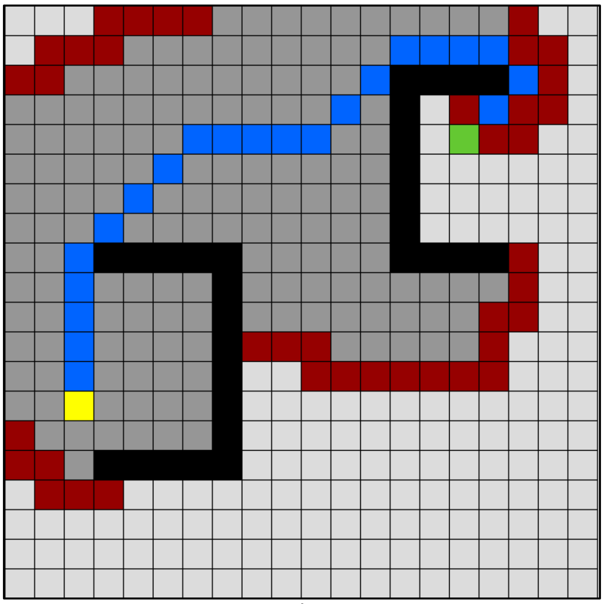

Fig 3: Traditional A* algorithm grid map

It can be seen from Figure 3 that the path connected by blue grid lines between the starting point and the end point is the path found by the traditional A* algorithm. The traditional A* algorithm has many problems such as traversing many useless points and poor smoothness of turning.

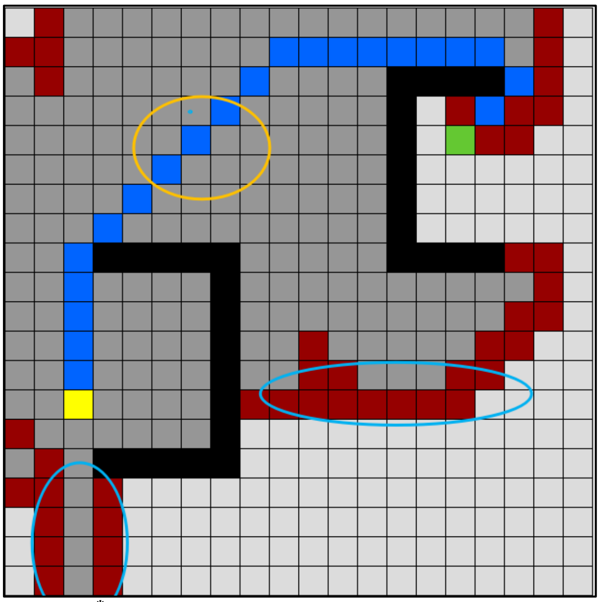

Fig 4: A* algorithm for modifying heuristic function

It can be seen from Figure 3 and Figure 4 that in the improved A* algorithm, the search nodes of the traditional A* algorithm in Figure 3 are reduced at the yellow circle, which is achieved by modifying the heuristic function of A*.However, there are many additional exploration points in the two blue circles, which need to be further improved. Path planning should not only consider shortening the planning time, but also consider the smoothness of the path. Therefore, it is necessary to increase the search points to reduce the number of turns. For reducing the number of search points, a streamlined strategy can be used.

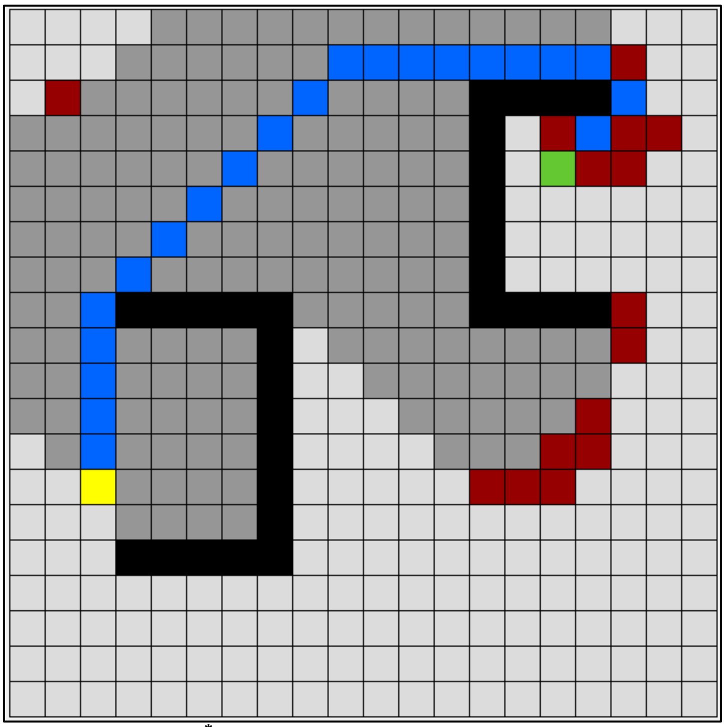

Fig 5: A* algorithm with streamline strategy

From the comparison between Figure 5 and Figure 4, it can be seen that the A* algorithm with the streamlined strategy needs to search significantly less paths. The improved A* algorithm has 170 search points. Compared with the 193 points of the A* algorithm modified by the heuristic function, it reduces 23 points and optimizes 11%.The number of turns is reduced from 6 to 4 in the traditional A* algorithm, which increases the smoothness of the path and reduces the time required for AGV to turn in the real environment.

4. CONCLUSION

Aiming at the problems of low search efficiency, too many turning points and unequal paths in the traditional A* algorithm, this paper improves the traditional A* algorithm by modifying the heuristic function and simplifying the search strategy. The heuristic function is modified to reduce the emergence of useless search points and improve the search efficiency. Adopt a streamlined search strategy to reduce the emergence of useless search points and improve search efficiency. The simulation results show that the number of search points and the number of path turns are optimized, and the improved A* algorithm is better than the traditional A* algorithm. The next research will start with the evaluation function of the A* algorithm to further improve the search efficiency of the A* algorithm and shorten the search time.

REFERENCES

1. Bina. (2022). Research on intelligent manufacturing assembly line workshop layout considering AGV path planning [J]. *Journal of Zhejiang University of Technology*, 50(5), 568-573.
2. Wang, H., & Wang, L. (2023). Review of robot path planning algorithms [J/OL]. *Journal of Guilin University of Technology*, 1-15[2023-02-23].
3. Chen, Y., Zhuang, L., & Zhu, L. (2017). Research on path planning of parking system based on improved Dijkstra algorithm [J]. *Modern manufacturing engineering*, (8), 63-67.
4. Song, Y., & Wang, Z. (2019). Mobile Robot Path Planning Based on Improved Genetic Algorithm [J]. *Modern Electronic Technology*, 42(24), 172-175.
5. Xie, C., Gao, S., & Sun, X. (2022). A path planning algorithm combining improved A* algorithm and Bessel curve optimization [J]. *Journal of Chongqing University of Technology (Natural Science)*, 36(7), 177-187.
6. Wang, Z., Zeng, G., & Huang, B. (2020). Mobile robot path planning algorithm based on improved bidirectional A* [J]. *Sensors and microsystems*,

39(11), 141-143 + 147. DOI:10.13873/J.1000-9787(2020)11-0141-03.

7. Li, X., Xiong H., & Tao, Y. (2021). Global optimal path planning for mobile robots based on improved A* algorithm [J]. *High technology communication*, 31(3), 306-314.
8. Wang, H., Ma, Y., & Xie, Y. (2010). Mobile robot path planning based on smooth A * algorithm [J]. *Journal of Tongji University (Natural Science Edition)*, 38(11), 1647-1650, 1655.
9. Qingji, G., Yongsheng, Y., & Dandan, H. (2005). Feasible path search and optimization Based on an improved A* algorithm. *China Civil Aviation College Journal*, *23*(4), 42-44.
10. Wu, P., Sang, C. J., Lu, Z. H., Yu, S., Fang, L. Y., & Zhang, Y. (2019). Research on mobile robot path planning based on improved A* algorithm [J]. *Computer Engineering and Applications*, *55*(21), 227-233.
11. Gu, C. (2014). Application of improved A* algorithm in robot path planning [J]. *Electronic Design Engineering*, 22(19), 96-98,102.