

# Application of A\* algorithm in intelligent vehicle path planning

Ruili Wang<sup>1</sup>, Zhizhan Lu<sup>2</sup>, Yunfeng Jin<sup>3</sup>, Chao Liang<sup>4</sup>

School of Civil Engineering and Transportation, Beihua University, Jilin, China

<sup>4</sup>Corresponding author

E-mail: <sup>1</sup>[Krystalpolly@163.com](mailto:Krystalpolly@163.com), <sup>2</sup>[290999487@qq.com](mailto:290999487@qq.com), <sup>3</sup>[324656658@qq.com](mailto:324656658@qq.com), <sup>4</sup>[4398503241@qq.com](mailto:4398503241@qq.com)

Received 25 June 2022; received in revised form 15 July 2022; accepted 26 July 2022

DOI <https://doi.org/10.21595/mme.2022.22828>



Copyright © 2022 Ruili Wang, et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract.** Path planning is one of the important directions in the field of intelligent vehicles research. Traditional path planning algorithms generally use Dijkstra algorithm, Breadth-First-Search (BFS) algorithm and A\* algorithm. Dijkstra algorithm is a search-based algorithm, which can search to an optimal path, but the disadvantage is too many expansion nodes, which leads to insufficient search efficiency. BFS algorithm is a heuristic search algorithm, which reduces the disadvantage of too many expansion nodes and improves the search efficiency by heuristic function. A\* algorithm is a heuristic search algorithm that combines Dijkstra's algorithm and BFS algorithm, which has higher search efficiency and can search to an optimal path at the same time, but it is still lacking in the search mode and smoothness of the planned route. This paper first introduces the general path planning algorithm, then introduces and analyzes the A\* algorithm, and proposes improvement measures for its shortcomings; finally, the executability and effectiveness of the improved algorithm are tested using simulation, and compared with the traditional A\* algorithm, and the results show that the improved A\* algorithm has good effect on path planning of intelligent vehicles.

**Keywords:** path planning, A\* algorithm, intelligent vehicles, heuristic function.

## 1. Introduction

In recent years, with the rapid development of artificial intelligence technology and cloud computing, the field of Intelligent Vehicles research has also risen to a new level. Among them, path planning is one of the key technologies in the field of Intelligent Vehicles research, which refers to finding an optimal path from the start position to the target position in a real road environment for smart cars according to set performance indicators (such as optimal distance time, no collision, etc.) [1]. There are many kinds of path planning algorithms, and the more common ones used for path planning of smart cars are Dijkstra algorithm, BFS algorithm, A\* algorithm [2]. In which, Dijkstra algorithm can find an optimal path, the shortcoming is that the search efficiency is too low. BFS algorithm compared with Dijkstra algorithm, the search efficiency is improved, but it is not sure whether an optimal path can be found. A\* algorithm is a heuristic search algorithm, combining the advantages of the first two functions, the calculation is simple and fast, easy to operate, and is widely used in two-dimensional path planning design, but it still has some shortcomings [3]. Therefore, in this paper, we first introduce the general pathfinding algorithm, then analyze the A\* algorithm and improve and optimize in the direction of deficiencies. The results of the optimization are verified by MATLAB simulation, and finally the paper is concluded.

## 2. Traditional path planning algorithms

### 2.1. Dijkstra algorithm

Dijkstra algorithm was proposed by Johnson in 1973 [4]. Dijkstra algorithm is mostly used for solving the shortest distance from one vertex to other vertices in the entitled graph. The algorithm



to ensure that the intelligent vehicle obtains the optimal path while maintaining the efficiency of path search as much as possible. The core part of the algorithm lies in the design of the estimation function [6], and the general form of the estimation function of the A\* algorithm is Eq. (1):

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

where  $f(n)$  is the cost estimate from the initial state to the target state via state  $n$ ,  $g(n)$  is the actual cost from the initial state to state  $n$ , and  $h(n)$  is the estimated cost of the best path from state  $n$  to the target state, also known as the heuristic function.

If the value of  $h(n)$  is low, the algorithm can obtain the shortest path, but the search operation is slow. If the value of  $h(n)$  is high, the search speed of A\* algorithm will become faster, but the obtained path is not the optimal path [7]. Manhattan distance, Euclidean distance, Chebyshev distance, etc. are often used as heuristic functions in practical calculations [8]. For any two points A and B in a two-dimensional right-angle coordinate system the coordinates of the two points are  $(X_A, Y_A)$ ,  $(Y_B, Y_B)$  then the three distance functions are.

(1) Manhattan distance is the sum of the absolute value of the difference between the  $X$  and  $Y$  coordinates of two points A and B located in the coordinate system:

$$L_{AB} = |X_A - X_B| + |Y_A - Y_B|. \quad (2)$$

(2) Chebyshev distance is a metric in vector space, which is the maximum value obtained by comparing the difference between the horizontal and vertical coordinates between two points A and B, respectively, and then taking the absolute value, i.e.:

$$L_{AB} = \max\{|X_A - X_B|, |Y_A - Y_B|\}. \quad (3)$$

(3) Euclidean distance is the shortest linear distance between two points A and B located in the coordinate system, i.e.:

$$L_{AB} = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}. \quad (4)$$

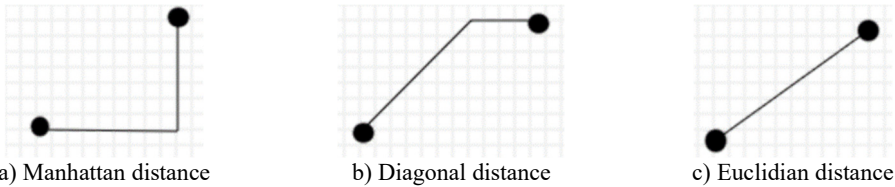


Fig. 3. Three distance diagram

A detailed representation of the three distances is given in Fig. 3.

Compare by three distances, Manhattan distance is simple to calculate and efficient in path planning, but the quality of the planned path is poor; Euclidean distance is the shortest and easy to get the shortest path, but the calculation is complicated leading to low solution efficiency; diagonal distance as a heuristic function is relatively simple to calculate and the valuation error of path planning is small, so the diagonal distance heuristic function is used in this paper.

The A\* algorithm is to evaluate each node by the estimation function to get the optimal node, and repeat the process until the target node is searched. The algorithm steps are as follows.

1) Initialize the program and generate the environment map boundary, starting and target points and obstacles.

2) Add the starting point S to the OPEN table.

3) Judge whether the OPEN table is empty, if it is empty, the algorithm is finished and the path cannot be found, otherwise go to the next step.

4) Find the node  $N$  with the smallest  $f(n)$  from the OPEN table, add it to the CLOSE table, and judge whether the node is the target node  $E$ . If it is the target node  $E$ , the algorithm ends, otherwise go to the next step.

5) Find all neighboring nodes  $X_i$  of node  $N$  ( $i < k$ ,  $k$  is the number of all nodes adjacent to node  $N$ ). If  $X_i$  is not in the OPEN table nor in the CLOSE table, calculate and add it to the OPEN table; if  $X_i$  is in the OPEN table, but smaller than the estimated value in the OPEN table, update the estimated value in the OPEN table and sort it from smallest to largest; if  $X_i$  is in the CLOSE table, ignore the point.

6) Over-cycle the algorithm until the target point  $E$  is in the CLOSE table, indicating that the optimal path has been found, or the OPEN table is empty, and the algorithm ends.

After running this algorithm, the target node will point to its leading node, with so on, until the leading of a node is the starting node, and the global optimal path can be obtained.

### 3.2. Deficiencies of the A\* algorithm

The analysis above shows that although the A\* algorithm is simple in principle and convenient to use, however, the algorithm needs to continuously update the OPEN array list and CLOSE array list during the search process to calculate the estimated value of each node, thereby finding the minimum node of the optimal path, thus also revealing that the A\* algorithm has certain deficiencies.

(1) the impact of the heuristic function on the algorithm. Heuristic function in the A\* algorithm, play a guiding role in the path search, the effect of its role is good or bad, directly determine the efficiency of the path search, but the traditional heuristic function will have some inefficiency, which leads to the efficiency of the path search is not high.

(2) There are too many unnecessary inflection points in the planned path, which leads to poor smoothness and increased danger, which is not conducive to smart cars driving in the actual environment.

## 4. A\* algorithm improvements

In this paper, we analyze the shortcomings of the above A\* algorithm in the search process and propose to improve the traditional A\* algorithm.

### 4.1. Optimization of the heuristic function $h(n)$

The traditional heuristic function will have some problems of low efficiency, in order to address this problem, this paper introduces the weight coefficient  $w(n)$  on the basis of the original heuristic function, and the improved heuristic function:

$$f(n) = g(n) + h(n) * w(n). \quad (5)$$

In Eq. (5),  $w(n)$  is the weight coefficient of the heuristic function  $h(n)$ . Adding this weight coefficient can make the search of the heuristic function more efficient. When it is far from the target point, the weight coefficient of  $w(n)$  is larger, which can accelerate the algorithm to search toward the target point, and when it is close to the target point, the weight coefficient of  $w(n)$  is smaller, which can make the algorithm search for a more accurate path. by using the optimized heuristic function can greatly improve the efficiency of intelligent vehicle path planning.

### 4.2. Path smoothing

When using the A\* algorithm for path search, there are many inflection points and redundant points because of the angle limitation. To address this problem, this paper uses the Floyd algorithm [9] to smooth the paths and remove the redundant inflection points, thus making the paths

smoother and also shortening the time loss of smart cars at the inflection points to improve the global path planning efficiency.

Floyd algorithm, also known as the interpolation method, is an algorithm for finding the shortest path between multiple source points in a given weighted graph, drawing on the idea of dynamic programming. Combining the A\* algorithm with the Floyd algorithm, the iterative optimization of the nodes by the Floyd algorithm results in an increase in the smoothness of the planned path compared to the path using only the A\* algorithm, while shortening the length of the path, which is beneficial to the global path planning of smart cars in the real environment. The schematic diagram of the Floyd algorithm [10] is shown in Fig. 4.

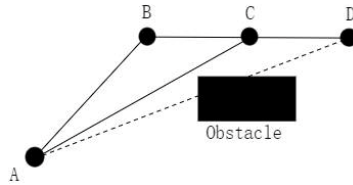


Fig. 4. Floyd algorithm schematic

$L(A, D)$  is the distance from node A to node D and  $R(A, D)$  is the path from node A to node D. From Fig. 4, it can be seen that there are obstacles between nodes A and D. Suppose  $L(A, D) = +\infty$ , the distance between nodes A and D is infinite.  $R(A, D) = A \rightarrow D$ , the path from node A to node D is node A pointing to node D.

Node B is the node planned to circumvent the obstacle between nodes A and D.

If:

$$L(A, B) + L(B, D) < L(A, D). \quad (6)$$

Then:

$$L(A, B) + L(B, D) < L(B, D), \quad (7)$$

$$R(A, D) = A \rightarrow B \rightarrow D. \quad (8)$$

Insert node C between nodes B and D.

If:

$$L(A, C) + L(C, D) < L(A, B) + L(B, D). \quad (9)$$

Then:

$$L(A, D) = L(A, C) + L(C, D), \quad (10)$$

$$R(A, D) = A \rightarrow C \rightarrow D. \quad (11)$$

The original planned path is optimized by Floyd's algorithm and updated from nodes A, B and D to the current nodes A, C and D. The turning angle of the path is significantly reduced, thus making the smoothness of the path planning significantly improved, which can ensure smoother operation of the intelligent vehicle.

## 5. Simulation verification

### 5.1. Creation of the simulation verification environment

The map modeling of the motion environment in which the intelligent vehicle is located is generally performed before the path planning. Depending on the type of map, the grid method,

topological map method and free space method are commonly used [11]. Grid method is commonly used by the A\* algorithm for path planning [12]. The basic idea of the grid method is to partition the environment region into a grid of the same size, each grid corresponds to a sub-partition of the real environment, and the grid map can characterize the initial and target locations of the smart car, the boundaries and obstacle areas within the environment region.

The environment modeling is based on MATLAB2021a platform. Firstly, the environment region is partitioned into  $N \times N$  equal partitioned grids, which are described in MATLAB. An example of a  $30 \times 30$  grid environment model built with MATLAB is shown in Fig. 5. The obstacle area is filled with black squares, yellow squares represent the starting point, and blue squares represent the target point.

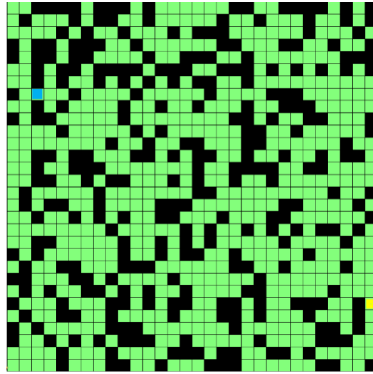


Fig. 5.  $30 \times 30$  grid map

## 5.2. Experimental analysis

In order to verify the effectiveness of the improved A\* algorithm, the simulation software platform MATLABR2021a is used to conduct simulation experiments and establish a raster environment map with a  $20 \times 20$  grid, set the proportion of two obstacles in the grid environment map to 20 % and 40 % respectively, and randomly generate the starting and ending coordinates. The path planning is performed in the grid map using the traditional A\* algorithm and the improved A\* algorithm.

Fig. 6 and Fig. 8 show the results of searching with the traditional A\* algorithm on the 20 % and 40 % random obstacle maps, respectively, and Fig. 7 and Fig. 9 show the results of searching with the improved A\* algorithm on the 20 % and 40 % obstacles, respectively.

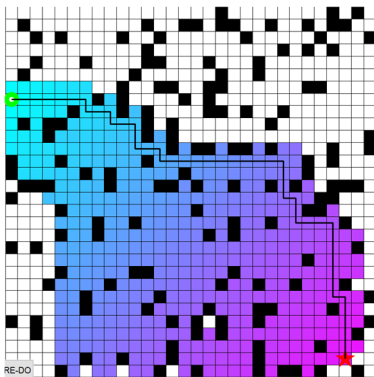


Fig. 6. 20 % barrier traditional A\* algorithm pathfinding results

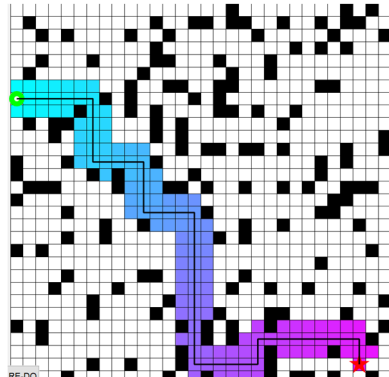


Fig. 7. 20 % barrier improvement A\* algorithm pathfinding results

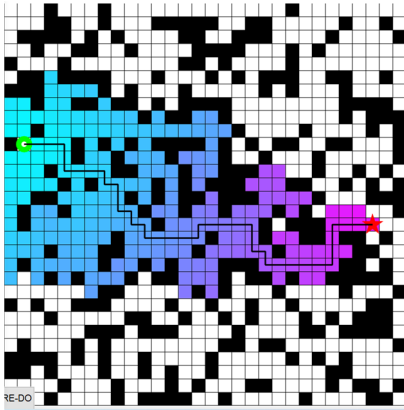


Fig. 8. 40% barrier traditional A\* algorithm pathfinding results

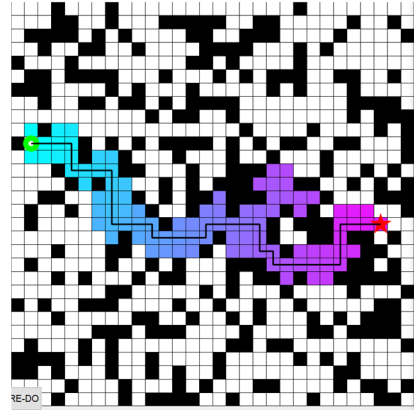


Fig. 9. 40% barrier improvement A\* algorithm pathfinding results

Comparing Fig. 6 and Fig. 7, Fig. 8 and Fig. 9, we can see that the paths are planned with the same starting point and target point in the obstacle map with the same occupancy ratio. It can be seen that the improved A\* algorithm has a significant decrease in the number of search nodes and a certain degree of reduction in the number of inflection points compared with the traditional A\* algorithm path finding results, so this paper improves the A\* algorithm, which has a good effect in reducing the number of search nodes and the number of inflection points.

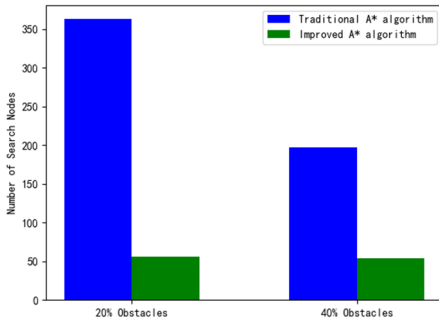


Fig. 10. Comparison of the number of search nodes

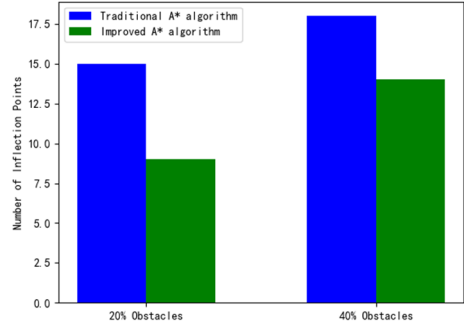


Fig. 11. Comparison of the number of inflection points

In order to more intuitively show the advantages of the improved A\* algorithm in the entire path planning, the number of search nodes and the number of inflection points in the search path are reduced, and the results are shown in Fig. 10 and Fig. 11.

From Fig. 10 and Fig. 11, it can be seen that the improved A\* algorithm proposed in this paper has a significant reduction in the number of search nodes and the number of inflection points compared with the traditional A\* algorithm. Therefore, the improved A\* algorithm of this paper for path planning of smart cars has good results in both reducing the number of search nodes and reducing the number of turns in the search path.

## 6. Conclusions

In this paper, we study the A\* algorithm in path planning of intelligent vehicles, firstly, we introduce traditional path planning algorithms and A\* algorithm, and address the problems such as the excessive number of expansion nodes and inflection points when A\* algorithm is used for path planning of intelligent vehicles. In this paper, an improved A\* algorithm is proposed, and the simulation results show that the optimization of the heuristic function reduces the number of

search nodes, thus improving the path search efficiency, and the Floyd algorithm is used to optimize the number of inflection points, which reduces the number of turns when searching the path and improves the smoothness of the search path.

## References

- [1] M. Zamirian, A. V. Kamyad, and M. H. Farahi, "A novel algorithm for solving optimal path planning problems based on parametrization method and fuzzy aggregation," *Physics Letters A*, Vol. 373, No. 38, pp. 3439–3449, Sep. 2009, <https://doi.org/10.1016/j.physleta.2009.07.018>
- [2] Q. Wu et al., "Real-time dynamic path planning of mobile robots: a novel hybrid heuristic optimization algorithm," *Sensors*, Vol. 20, No. 1, p. 188, Dec. 2019, <https://doi.org/10.3390/s20010188>
- [3] Z. Nie and H. Zhao, "Research on robot path planning based on Dijkstra and ant colony optimization," in *2019 4th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pp. 222–226, Nov. 2019, <https://doi.org/10.1109/iciibms46890.2019.8991502>
- [4] G. Qing, Z. Zheng, and X. Yue, "Path-planning of automated guided vehicle based on improved Dijkstra algorithm," in *2017 29th Chinese Control and Decision Conference (CCDC)*, pp. 7138–7143, May 2017, <https://doi.org/10.1109/ccdc.2017.7978471>
- [5] B. Awerbuch and R. G. Gallager, "Distributed BFS algorithms," in *26th Annual Symposium on Foundations of Computer Science (SFSC 1985)*, pp. 250–256, Oct. 1985, <https://doi.org/10.1109/sfcs.1985.20>
- [6] J. Wangdian, "Indoor mobile robot path planning based on improved A\* algorithm," (in Chinese), *Journal of Tsinghua University: Natural Science Edition*, Vol. 52, No. 8, pp. 5–10, 2012.
- [7] Z. Xu, X. Liu, and Q. Chen, "Application of improved Astar algorithm in global path planning of unmanned vehicles," in *2019 Chinese Automation Congress (CAC)*, pp. 2075–2080, Nov. 2019, <https://doi.org/10.1109/cac48633.2019.8996720>
- [8] R. Xuan, Z. Li, and Y. Wang, "Path planning of intelligent vehicle based on optimized A algorithm," in *2018 Chinese Automation Congress (CAC)*, pp. 526–531, Nov. 2018, <https://doi.org/10.1109/cac.2018.8623303>
- [9] S. Weiren and K. Wang, "Research on shortest path planning for mobile robots based on Floyd's algorithm," (in Chinese), *Journal of Instrumentation*, Vol. 30, No. 10, pp. 2088–2092, 2009.
- [10] H. Chen and R. Wangzhi, "Research on path planning of mobile robot based on A\* and Floyd algorithm," (in Chinese), *Construction Machinery Technology and Management*, Vol. 31, No. 3, pp. 43–45, Nov. 2018.
- [11] C. Wang et al., "Path planning of automated guided vehicles based on improved A-Star algorithm," in *2015 IEEE International Conference on Information and Automation (ICIA)*, pp. 2071–2076, Aug. 2015, <https://doi.org/10.1109/icinfo.2015.7279630>
- [12] J. Wuya and D. Yang, "Robot path planning based on a\* algorithm," (in Chinese), *Electronic Science and Technology*, Vol. 30, No. 6, pp. 124–127, 2017.

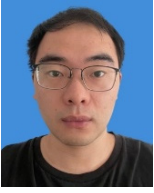


**Wang Ruili** is a graduate student at School of Civil Engineering and Transportation, Beihua University, China. His main research field is Intelligent vehicle path planning.



**Lu Zhizhan** is a graduate student at School of Civil Engineering and Transportation, Beihua University, China. His main research field is Intelligent vehicle map building and tracing.





**Jin Yunfeng** is a graduate student at School of Civil Engineering and Transportation, Beihua University, China. His main research field is Infrared and visible light image fusion.



**Liang Chao** received his Ph.D. and is currently a Professor in the Department of Vehicle and Civil Engineering at Beihua University in Jilin, China. His main research areas include vehicle operation quality and safety, and intelligent transportation. He has participated in several provincial projects, published more than 10 papers, and obtained several invention patents.