An Improved Path Planning Algorithm Based on A* Algorithm

Dongdong Li(✉), Xiaohou Shi, and Meiling Dai

China Telecom Corporation Research Institute Beijing, Beijing, China
{lidd4,daiml1}@chinatelecom.com

Abstract. Path planning is a crucial aspect of vehicle navigation, and this paper presents an enhancement to the classic A* algorithm to address key challenges in this domain. The proposed method aims to improve both the efficiency and safety of path planning. In practical applications, path planning encounters various issues, such as an excessive number of unnecessary nodes during the search process, resulting in suboptimal planning efficiency. Additionally, obstacles may be present along the route between the starting point and the target node, requiring obstacle path search. Moreover, traditional cost functions often fail to fully account for vehicle safety, thereby increasing the risk of collisions. To overcome these challenges, the proposed method incorporates two key enhancements. Firstly, it employs a node marking technique on the grid map to identify key nodes and reduce the search process for unnecessary nodes, thus enhancing planning efficiency. Secondly, an incremental expansion of search nodes is utilized, employing an improved A* algorithm with a modified cost function. This enables the algorithm to plan collision-free paths from the starting point to the key nodes while considering the distance cost associated with potential collisions, thereby enhancing vehicle safety. Experimental results demonstrate that the proposed method significantly enhances path planning efficiency by reducing the search efforts for unnecessary nodes, resulting in accelerated path planning. Furthermore, the improved cost function enables the generation of safe and feasible paths, thereby enhancing vehicle safety.

Keywords: Path planning · A* algorithm · Vehicle navigation

1 Introduction

Path planning is a fundamental technology widely employed in the transportation field, playing a pivotal role in optimizing traffic flow, alleviating congestion, enhancing efficiency, and improving traffic safety. As urbanization accelerates and traffic demands grow, the need for efficient route planning becomes increasingly crucial to enable individuals to reach their destinations quickly and safely. This study aims to investigate enhancements to existing path planning methods, addressing limitations in terms of efficiency and safety.

Y. Zhang et al. (Eds.): CENet 2023, LNEE 1125, pp. 187–196, 2024.
https://doi.org/10.1007/978-981-99-9239-3_19

Considerable research efforts have been devoted to improving the accuracy and efficiency of path planning. The classic A* algorithm [1], as a heuristic search algorithm, is extensively employed in path planning tasks. By evaluating the cost function and heuristic estimation function of nodes, the algorithm searches for the optimal path. However, when applied to large-scale maps and complex road networks, the A* algorithm encounters challenges such as excessive node searches and high computational complexity, leading to diminished efficiency.

To overcome these limitations, several studies have proposed enhanced methods. These methods incorporate techniques such as heuristic search strategies, pruning techniques, and map preprocessing [2] to improve path planning efficiency. Other approaches focus on vehicle safety and introduce technologies like collision risk assessment and traffic flow prediction [3] to plan safer and more reliable routes.

Nonetheless, these research methods possess certain drawbacks. Some enhancement methods may sacrifice path accuracy or struggle to handle complex traffic scenarios while improving efficiency. Although other methods prioritize security, they face challenges related to computational complexity and real-time performance, impeding their widespread practical implementation.

This paper proposes an improved path planning method that overcomes the limitations of traditional approaches while comprehensively considering both efficiency and safety aspects. The research methodology encompasses the following aspects: during node search, only the key nodes marked on the map are explored, reducing the search process for unnecessary nodes and improving path planning efficiency. Additionally, the cost function of the A* algorithm is enhanced by incorporating collision distance cost to enhance vehicle behavior.

2 Introduction to A* Algorithm

The A* algorithm, a renowned heuristic search algorithm, holds a significant position in the realm of path planning. By examining the node's cost function and heuristic estimation function, it efficiently determines the optimal path.

The fundamental principle underlying the A* algorithm involves a comprehensive consideration of both the actual cost of the node and the heuristic estimated value. This evaluation leads to the selection of the node with the minimum comprehensive cost for further exploration. Each node possesses a cost value that reflects the expense incurred along the path from the origin to that particular node. Typically, this value is expressed using a distance metric. Concurrently, the heuristic estimation function assesses the anticipated cost from the current node to the target node. Consequently, the A* algorithm assigns priority to nodes with lower comprehensive costs, facilitating the progression towards the ultimate goal during the search process.

During the planning process, the vehicle initiates its journey from the starting point and eventually reaches the target point by traversing an extended cycle. The specific steps of the conventional A* algorithm are depicted in Fig. 1.

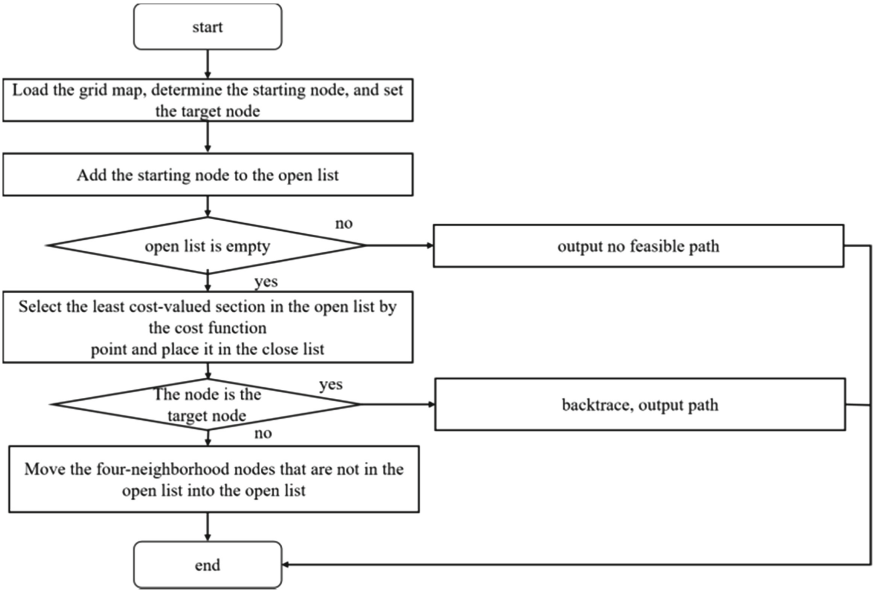

start
Load the grid map, determine the starting node, and set the target node
Add the starting node to the open list
open list is empty
no
output no feasible path
yes
Select the least cost-valued section in the open list by the cost function point and place it in the close list
The node is the target node
yes
backtrace, output path
no
Move the four-neighborhood nodes that are not in the open list into the open list
end

Fig. 1. Traditional A* algorithm process

3 Improved A* Algorithm

The proposed method begins by identifying essential nodes on the rasterized map. It then determines the target vector SG, which represents the connection line from the current node S to the target point G. Next, the method selects the key node N located closest to the target point G within the Manhattan distance around SG. In the event that obstacle nodes are present in the search area of N, the method applies an incremental expansion of the A* algorithm along with an enhanced cost function to search for a path from S to N that circumvents obstacles. This process is repeated iteratively, selecting key nodes N until N corresponds to the target point G, thus concluding the path planning phase. A visual representation of the overall process can be observed in Fig. 2.

The initial step involves rasterizing the map and subsequently identifying key nodes within the map. Marked path turning nodes and path dead-end nodes are added to the list of path key nodes. The process of identifying these key nodes is depicted in Fig. 3.

The next step involves generating the target vector SG, which extends from the current node S (initially set as the starting point) to the target node G. The target vector SG is represented in Fig. 4, providing a visual representation of its direction and orientation.

In Fig. 4, the key nodes A, B, C, and D are positioned around the target vector SG. Using the SG vector as a reference, the key node N closest to the target point G is selected. To determine this, the Manhattan distance [4] between each key node and the target point G is calculated, and the key node with the smallest distance is chosen. Taking Fig. 4 as an example, the distance between the current node S and the target point G can be computed as follows:

$$\mathrm{dM} = |\mathrm{Gx} - \mathrm{Sx}| + |\mathrm{Gy} - \mathrm{Sy}| \tag{1}$$

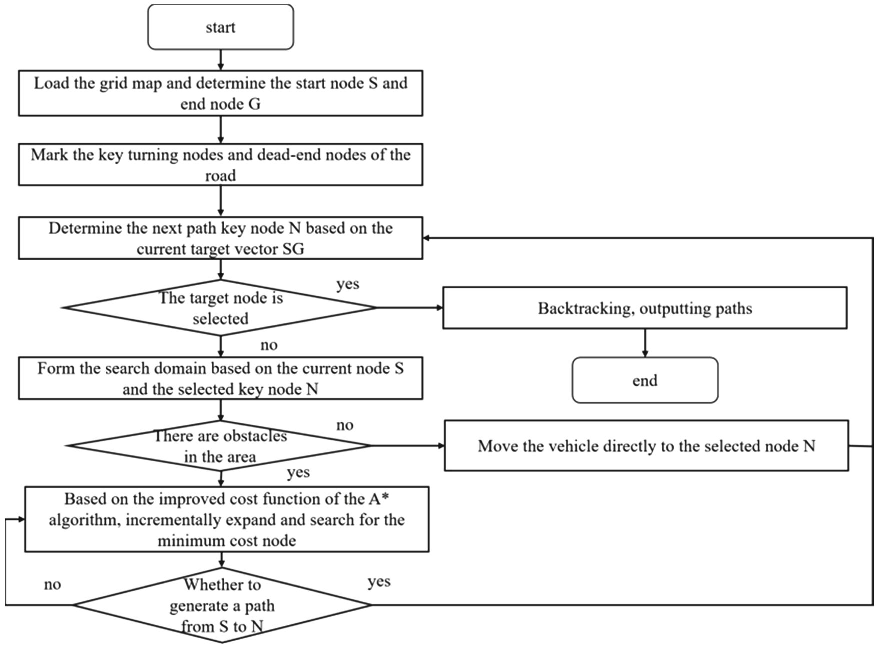

start
Load the grid map and determine the start node S and end node G
Mark the key turning nodes and dead-end nodes of the road
Determine the next path key node N based on the current target vector SG
The target node is selected
yes
Backtracking, outputting paths
end
no
Form the search domain based on the current node S and the selected key node N
There are obstacles in the area
no
Move the vehicle directly to the selected node N
yes
Based on the improved cost function of the A* algorithm, incrementally expand and search for the minimum cost node
no
Whether to generate a path from S to N
yes

Fig. 2. The overall flow chart of the path planning algorithm

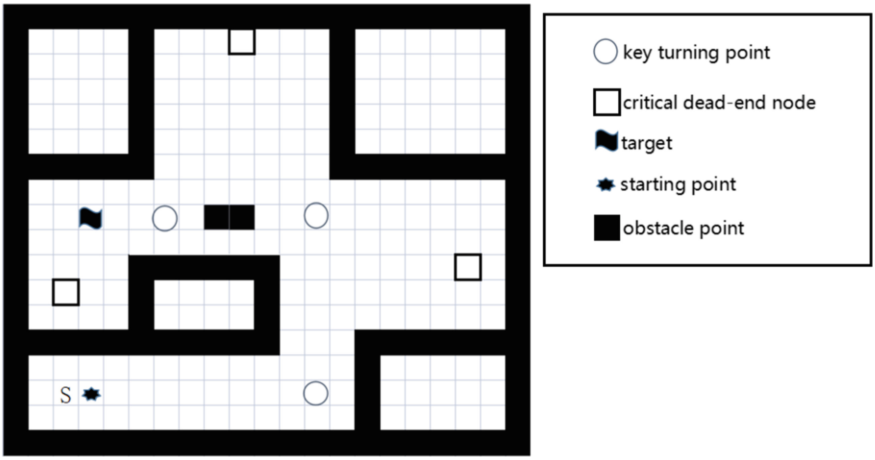


Fig. 3. Raster map with key nodes

The Manhattan distance dM represents the distance between node S and node G. The coordinates of node S are denoted as (Sx, Sy), while the coordinates of node G are denoted as (Gx, Gy). Upon identifying the key node N with the smallest distance to the target point G, an assessment is made to determine if N is a path dead-end node. If it is indeed a dead-end node, a re-selection of the key node N is required. Conversely, if N is

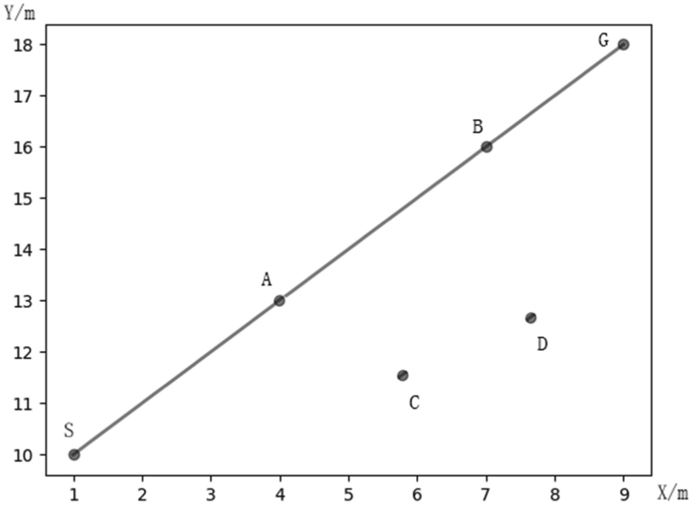


Fig. 4. Key node selection strategy based on target vector

not a dead-end node, the algorithm proceeds to evaluate the existence of obstacle points within the search area of the current node S and node N.

Referring to Fig. 5, the search area encompasses both node S and node N, with the length of the search area being |Nx − Sx| and the width corresponding to the road length of 7 m. Here, Nx represents the x-axis coordinate of the selected node N. If no obstacle nodes are present within the search area, the vehicle can directly navigate to node N. However, in the presence of an obstacle point, the A* algorithm utilizes an incremental expansion technique to plan an obstacle-free path leading to node N. During the search process, the improved cost function, as outlined in step C, is employed for node evaluation and selection.

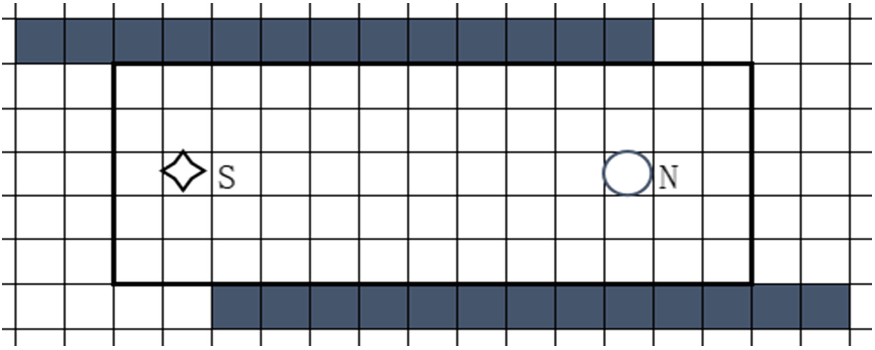


Fig. 5. Obstacle detection area indication

C. This method divides the obstacles in the search area into two types. The first type is obstacles with small size and high probability of movement, such as stationary pedestrians beside the road, with a length l ∈ (1, 2) m. the second type is Obstacles with low probability of movement but large size, such as parking next to the road, have length l ∈ (3, 4) m.

Based on the vehicle speed, safe distance between vehicles and a large number of simulation test analysis, an anti-collision safety distance model is designed, in which the safety distance between the vehicle and obstacles is related to the vehicle speed and road friction coefficient, and the calculation formula is as follows:

$$\mathrm{ds} = \mathrm{kv}^2/(2\mu \mathrm{g}) + (2 - \mu)\mathrm{du} \tag{2}$$

Among them, ds is the safe distance between the vehicle and the obstacle, k is the weight of different types of obstacles, when the obstacle is large, k = 1.5, and when the obstacle is small, k = 1.2. v is the vehicle speed. μ is the vehicle The friction coefficient of the ground. g is the acceleration of gravity, the value is 10 m/S2. du is the unit distance, the value is 1 m.

Figure 6 shows the relationship model diagram of safe collision distance, road friction coefficient, and vehicle speed. This relationship model diagram is based on simulation tests of a large number of vehicle collision scenarios., the safe collision distance gradually increases, which is also in line with the driver's collision avoidance habits during vehicle driving in the real world.

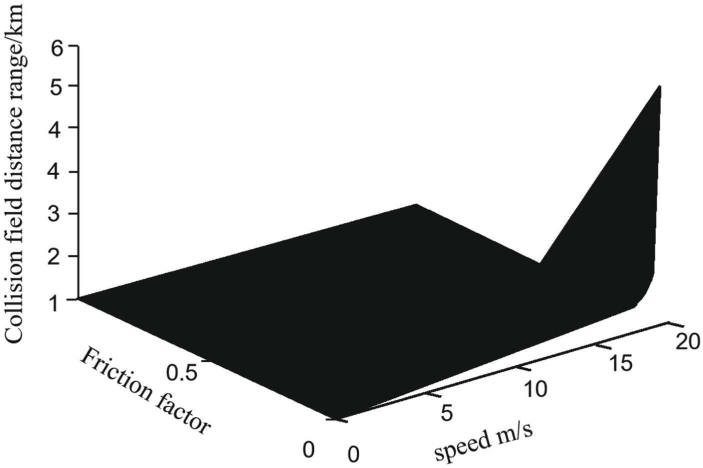
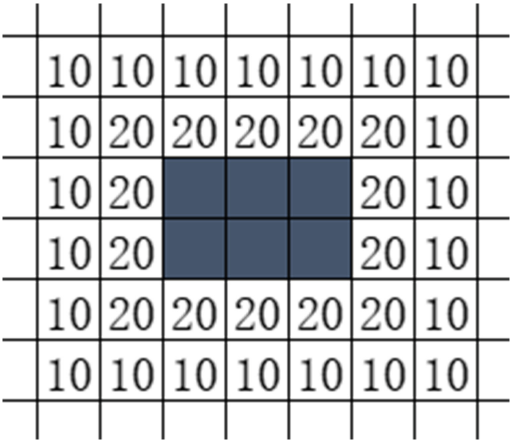


Fig. 6. Obstacle collision field model

The total number of layers c of the expanded grid can be obtained from ds as:

$$\mathrm{c} = \lceil ds/10 \rceil \tag{3}$$

In the formula, ⌈ ⌉ means round up. Re-assign the distance of the collision field to 10 m × c, and use it as the cost value of the innermost grid of the extended grid, and then carry out decremental assignment in multiples of 10 m. If the extended grid of obstacles contains obstacle grids, It is not assigned a value. As shown in Fig. 6, when the vehicle speed is 15 m/s and the road surface adhesion coefficient is 0.8, ds = 18.075 m can be obtained from formula (3), and c = 2 can be obtained from formula (3), so directly o(N) = 20 m is assigned to the first layer grid outside the obstacle boundary, and the cost value o(N) = 10 m is assigned to the second layer grid outside the obstacle boundary (Fig. 7).



10	10	10	10	10	10	10
10	20	20	20	20	20	10
10	20	■	■	■	20	10
10	20	■	■	■	20	10
10	20	20	20	20	20	10
10	10	10	10	10	10	10

Fig. 7. Extend grid based on collision field distance

The overall cost function, the improved A* algorithm cost function is:

$$f(N) = g(N) + k1h(N) + k2o(N) \tag{4}$$

In formula (4), f(N) = g(N) + h(N) is the cost function of the original A* algorithm, o(N) is the collision field distance cost of different obstacles. k1 and k2 are different costs The weight of the function. Among them, the larger k1 is, the faster the path tends to the target point, and the shorter the calculation time is, but this will affect the optimality of the path, while the value of k2 will directly affect the distance between the planned path and the obstacle boundary, and then affect the driving of the vehicle safety. After many simulations and verifications, this paper takes k1 = 2 and k2 = 2. For example, the A* algorithm gradually selects the key nodes with the smallest replacement value through repeated iterations, thereby generating an obstacle-free planning path from S to N. If N is not the target point, continue to execute step B, otherwise end the path planning.

4 Experiment and Analysis

To evaluate the performance of the improved A* algorithm, this study compares it with the traditional A* algorithm and the Weighted-A* algorithm on a regional grid map derived from real-world data. To validate the generalization ability and enhanced performance of the proposed algorithm, a simulation-based comparative analysis is conducted using a real scene map.

In this research, the same starting point and target point are set for all three algorithms, as depicted in Fig. 8. The grid map size is standardized at 30 × 30 m. Various parameters, including path length, average computation time, number of expanded nodes, number of turns, and minimum distance from road boundaries or other obstacle boundaries, are considered. It should be noted that the A* algorithm is classified as an optimal search algorithm, resulting in a unique output path. However, due to the influence of system test environment and hardware performance, the computation time may vary within a single iteration, leading to differences in time consumption across different test scenarios. To mitigate this variability, the algorithms are tested 50 times under identical conditions, and the average computation time for each algorithm is calculated, ensuring a comprehensive evaluation of the improved algorithm's computational efficiency. The final planning result for a specific scenario is presented in Fig. 8.

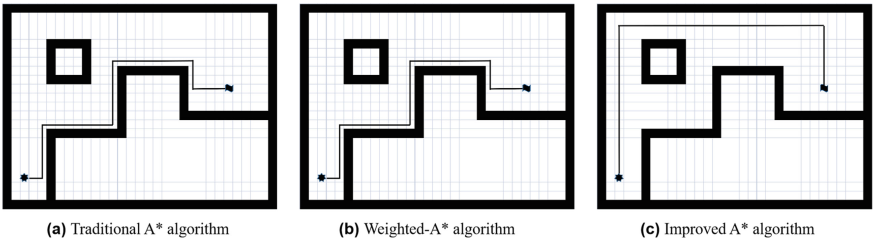

(a) Traditional A* algorithm (b) Weighted-A* algorithm (c) Improved A* algorithm

Fig. 8. Driving coverage trajectory of different algorithms under the real scene map

Figure 8 illustrates that the path results obtained by the traditional A* algorithm and the Weighted-A* algorithm exhibit similarity. However, their trajectories tend to hug the road edges with numerous turns, rendering them unsuitable for unmanned vehicles in this particular scene. In contrast, the path generated by the proposed improved algorithm not only maintains a safe distance from obstacles but also prioritizes straight trajectories to ensure the safety and feasibility of the planned path for unmanned vehicles.

The time-consuming results from 50 test calculations are presented in Fig. 9. It can be observed that the improved A* algorithm demonstrates significantly shorter computation times compared to the traditional A* algorithm. However, its computation time is comparable to that of the Weighted-A* algorithm in the real map scene. Furthermore, the fluctuation range of computation times for different algorithms remains within 20 ms, highlighting the improved algorithm's stable calculation performance.

The main parameters obtained from 50 test runs for different algorithms are recorded and presented in Table 1.

From Table 1, the number of expanded nodes significantly reduces with the improved A* algorithm compared to the traditional A* algorithm and the Weighted-A* algorithm. The algorithm improvement demonstrates a moderate enhancement of 8.3%. Furthermore, when compared to the traditional A* algorithm and the Weighted-A* algorithm, the improved A* algorithm showcased a decreased number of turns, ensuring a sufficient minimum distance between the path and various obstacle boundaries to guarantee driving safety. In conclusion, the proposed algorithm's effectiveness in this study is notably prominent.

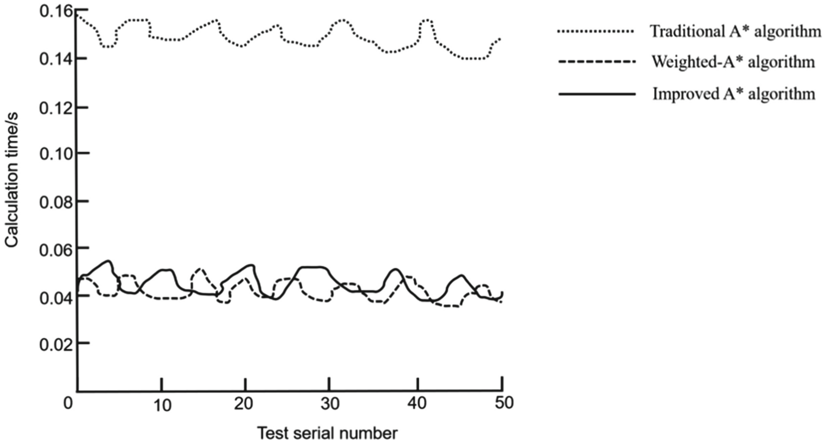


Fig. 9. Calculation time of different algorithms

Table 1. Test results of different algorithms.

Algorithm	Path length	Average time	Nodes number	Turns number
Traditional A* algorithm	58	102	913	8
Weighted-A* algorithm	58	30	325	8
Improved A* Algorithm	66	25	10	3

5 Conclusion

This study presents an improvement to the conventional A* algorithm for addressing the path planning problem, with the objective of enhancing both the efficiency of path planning and vehicle safety. The traditional A* algorithm encounters challenges, such as extensive node searches and high computational complexity, particularly when applied to large-scale maps and intricate road networks, resulting in suboptimal efficiency. To address these shortcomings, this paper proposes enhancements to the traditional A* algorithm by incorporating key node search techniques and introducing a collision field model based on a safe distance. Through experimental validation, the effectiveness and advantages of the proposed method are demonstrated, signifying its significance for practical applications. By leveraging key node identification and an improved cost function, the efficiency of path planning and the safety of vehicle navigation can be significantly improved. Nevertheless, there remain areas for further improvement and investigation, such as performance optimization when dealing with complex traffic scenarios and large-scale maps, as well as integration with real-world traffic environments. Advancements in these areas will contribute to enhancing the practicality and adaptability of path planning algorithms.

References

1. Hart, E., Nilsson J.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cyber. **4**(2), 100–107
2. Sturtevant, N.: Benchmarks for grid-based pathfinding. IEEE Trans. Comput. Intell. AI Games **4**(2), 144–148
3. Wei, Z., Xia, F., Liu, Y., Xu, Y.: Safe path planning algorithm based on collision risk assessment for intelligent vehicles. IEEE Trans. Intell. Transp. Syst. **17**(8), 2164–2173
4. Zhao, Z., Meng Z.: Path planning of service mobile robot based on adding- weight A* algorithm. J. Huazhong Univer. Sci. Technol. (Nat. Sci. Ed.) **36**(S1), 196–198 (2008)