Dynamic adaptive security path planning based on A* algorithm

Fan Lu, Cao Han*, Ganwenxiang Wu, Mingrui Lu, Jikun Yang, Baorui Miao and Yingjie Ma
School of Electrical Engineering Anhui Polytechnic University, Anhui, China

* corresponding author: hanchaozh@126.com

Abstract. Aiming at the problems of large memory consumption, long operation time and low security of traditional A* algorithm, a dynamic adaptive improved algorithm based on A* algorithm is proposed. Firstly, according to the heuristic function of the proportion of obstacles in the map, the adaptive dynamic weighting is carried out to shorten the search time. Secondly, the jump point search (JPS) algorithm is integrated to remove a large number of redundant points and improve the running speed of the algorithm. Finally, according to the relative position of the starting point and the target point, the obstacles are expanded dynamically to keep a safe distance between the planned path and the obstacles. In order to verify the effect, the improved algorithm is tested in different sizes and different proportion of obstacles. The results show that the improved algorithm is efficient and safe in global path planning.

1. Introduction
Mobile robot technology was a hot research field in recent years and the system was usually composed of environmental perception, decision making and planning, behavior control and execution and other modules [1]. In decision making and planning module, path planning problem has long been the attention of the researchers, and become one of the hot research areas.

Path planning is given a known or unknown environment, planning out a path from starting point to target point without collision safety, generally can be divided into the global path planning [2-4] and local path planning [5-7]. This paper mainly studies the global path planning based on A* algorithm. Dijkstra algorithm is a typical single-source shortest path algorithm. Although this algorithm can search the optimal path, its efficiency is very low. Reference [8] proposed the JPS algorithm, which reduced a large number of redundant points generated by the traditional A* algorithm and improved the running speed. However, the security of the path could not be guaranteed and the path might not be the optimal path. Reference [9] proposed the algorithm of static weighting and JPS fusion, which accelerated the search speed, but the path security performance was low. Reference [10] improved the heuristic function and shortened the path length, but the running time was longer. Reference [11] proposed the SD-JPS algorithm, which guaranteed the path security by introducing the node domain matrix, but increased the running time of the algorithm.

To sum up, this paper proposes an improved A* algorithm based on the traditional A* algorithm, which performs adaptive dynamic weighted processing according to the heuristic function of the proportion of the number of obstacles in the map. The obstacles are dynamically expanded and the JPS algorithm is fused to reduce redundant points, so a path with high safety performance can be obtained quickly. Finally, the risk assessment function is used to evaluate the security of the path and the experimental simulation results verify the security and reliability of the proposed algorithm.
2. **Introduction to the traditional A* algorithm**

The principle of the traditional A* algorithm is to continuously access the nodes surrounding the current point from the starting point, add the surrounding nodes to the set of nodes to be scanned, calculate generation value of the node and update the node with the smallest generation value that is the current node, and then repeat the previous process until the target node is found. Its cost function is shown in Eq. (1):

\[ f(n) = g(n) + h(n). \] (1)

Where, \( n \) represents the node currently evaluated, \( f(n) \) represents the total cost function of the current node, \( g(n) \) represents the cost function from the starting point to the current evaluation node, and \( h(n) \) represents the cost function from the current evaluation node to the target point.

In general, the number of nodes evaluated by the traditional A* algorithm is relatively large, and the overall efficiency is low. When planning the path, many redundant points will be generated.

3. **Improvement of A* algorithm**

3.1 **Dynamic adaptive weighting**

Since the heuristic function of the traditional A* algorithm has a significant impact on the number of nodes to be evaluated, this paper weights the exploratory function dynamically. When the current point is far from the target point, it moves faster to the target point. When it is close to the target point, it dynamically adjusts weighting, reduces job expedition and improves search range. The optimization process is as follows:

\( h(n) \) is dynamically weighted by exponential function, and the processed function is expressed by \( h(n) \), as shown in Eq. (2):

\[ H(n) = h(n) \cdot \exp[h(n)]. \] (2)

\( n \) is transformed by introducing function \( \alpha(m) \), as shown in Eq. (3):

\[
\begin{align*}
H(n, m) &= h(n) \cdot \exp[\alpha(m) \cdot h(n)] \\
\alpha(m) &= k \cdot (1 - m).
\end{align*}
\] (3)

Where, \( m \) represents the ratio of the obstacle area under the current grid map to the entire map area. \( k \) represents a constant, and the value in this paper is 0.006. The improved cost function is expressed by \( f(n, m) \), as shown in Eq. (4):

\[
\begin{align*}
f(n, m) &= g(n) + H(n, m) \\
H(n, m) &= h(n) \cdot \exp[\alpha(m) \cdot h(n)].
\end{align*}
\] (4)

When the value of \( m \) increases, the value of \( H(n, m) \) will become smaller and the enlightening of the function will decrease. When the value of \( m \) becomes smaller, that is, the proportion of obstacles on the map decreases, the heuristic function increases, and the search efficiency improves.

3.2 **Building an environment map**

In the research field of mobile robot, there are many methods of environment map construction, which can be roughly divided into the following types: geometric map, topological map, grid map and hybrid map. This paper uses this method to construct the map, as shown in Figure 1. The black networks represent obstacles and the white networks represent areas that can be crossed.

During the operation of mobile robot, there are usually 4 or 8 neighborhood moving directions. In order to ensure the precision and flexibility of mobile robot movement, this paper adopts the 8-neighborhood movement direction, as shown in Figure 2.
3.3 Search strategy

In the traditional A* algorithm, the number of nodes evaluated is influenced by the heuristic function $h(n)$. In order to reduce the time of path planning, this paper integrates JPS algorithm, reduces the number of evaluation nodes and search area, and improves the operation efficiency. The flow chart of the improved algorithm is shown in Figure 3.

![Flow chart of improved algorithm.](image)

(a) The starting and target points are in a straight line. (b) The starting and target points are not in a straight line.

**Figure 4.** Dynamic obstacle expansion.

The pentagonal grid in the Figure 4. denotes the starting point, with coordinate $(x_o, y_o)$. The triangular grid denotes the target point, with coordinate $(x_e, y_e)$. The black grids denote obstacles, and the coordinates are $(x_o, y_o)$. The gray grids denote the expansion. If $y_s = y_e$, as shown in Figure 4 (a), the grids at $(x_o - 1, y_o)$ and $(x_o + 1, y_o)$ are expanded. If $x_s = x_e$, then the grids at $(x_o, y_o - 1)$ and $(x_o, y_o + 1)$ are inflated. As shown in Figure 4 (b), $x_s - x_e < 0$ and $y_s - y_e < 0$, the grids at $(x_o - 1, y_o)$ and $(x_o, y_o - 1)$ are expanded. If $x_s - x_e < 0$ and $y_s - y_e > 0$, then the grids at $(x_o - 1, y_o)$ and $(x_o, y_o + 1)$ are expanded. Similarly, the other two directions are expanded in accordance with this method. The improved path basically keeps a safe distance from obstacles, which reduces the risk of collision and improves the safety performance.

In order to evaluate the risk degree of mobile robots colliding with obstacles, this paper adopts the two-dimensional Gaussian model [12] to establish a risk assessment function to evaluate the safety of the planned path. The function is shown in Eq. (5):
\[ W = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \delta(x, x_{\text{obs}}) \]
\[ \delta(x, x_{\text{obs}}) = \begin{cases} d^{\alpha} \delta^{\beta}, & \delta \leq d, \\ 0, & \delta > d \end{cases} \]

Where, \( W \) represents the risk coefficient, \( M \) represents the total length of the path, \( N \) represents the total number of obstacles within the influence range of the path, \( \delta(x, x_{\text{obs}}) \) represents the distance between each node in the path and the obstacles within the influence range of the node, and \( d \) represents the influence range of each node. The range representing the eight fields around the node. \( \alpha \) and \( \beta \) represent the material parameters of the mobile robot and the obstacle. In this paper, \( \alpha = 1, \beta = 2 \).

4. Simulation experiment and analysis

The simulation experiment in this paper is carried out on an Intel(R) Core (TM) i5-10400 six-core CPU and 8G RAM computer. In the process of experimental simulation, the algorithm running time, the number of nodes to be evaluated, the path security performance and the path length are compared. Taking the map size of \( 80 \times 80 \) and the obstacle proportion of 15% as an example, the path comparison between the improved algorithm and the traditional A* algorithm is shown in Figure 5.

(a) Improved algorithm for path planning. (b) Traditional A* path planning. Figure 5. Comparison chart of path planning.

The blue grid denotes the final generated path and is marked with red lines, the orange grid denotes the nodes added to the Closed List, the light gray grid denotes the nodes added to the Open List, and the dark gray grid denotes the nodes where the obstacles have expanded. The upper left grid is the starting point and the lower right grid is the target point. The traditional A* algorithm takes 99.83ms to plan the path, the path length is 102, the number of evaluated nodes is 3069, and the risk coefficient is 31.92. The path planning time of the improved algorithm is 8.72ms, the path length is 109, the number of evaluated nodes is 39, and the risk coefficient is 24.09. Although the path length of the improved algorithm is 6.86% higher than that of the traditional A* algorithm, the running time, the number of evaluated nodes and the risk coefficient are reduced by 91.27%, 98.73% and 24.53% respectively. The efficiency and security of the planned route is improved, so the planned route is better.

4.1 Running time

The comparison of the running time of the traditional A* algorithm, JPS algorithm and the improved algorithm is shown in Figure 6. The red, green and blue bar charts represent the map sizes of \( 20 \times 20 \), \( 40 \times 40 \) and \( 80 \times 80 \) respectively. In general, the running time of the improved algorithm in the three environments is lower than that of the traditional A* algorithm and JPS algorithm, especially with the growth of map size, the running efficiency of the improved algorithm is significantly improved.

Figure 6. Comparison chart of running time.
4.2 Risk coefficient
The risk coefficient comparison among the traditional A* algorithm, JPS algorithm and the improved algorithm is shown in Figure 7. The red, green and blue bar charts represent the map sizes of 20×20, 40×40 and 80×80 respectively. In general, as the map size increases, the risk coefficient of these three algorithms increases, but the improved algorithm is always lower than the traditional A* algorithm and JPS algorithm.

4.3 Evaluation nodes
The comparison of evaluation nodes of traditional algorithm, JPS algorithm and improved algorithm is shown in Figure 8. In the Figure, the red, green and blue bar charts represent the map sizes of 20×20, 40×40 and 80×80 respectively. On the whole, the number of evaluation nodes increases with the growth of map size, and the number of evaluation nodes of the improved algorithm is generally smaller than that of JPS algorithm, much smaller than that of the traditional A* algorithm, which greatly reduces the memory overhead.

4.4 Path length
The comparison of path length among the traditional algorithm, JPS algorithm and improved algorithm is shown in Figure 9. In the Figure, the red, green and blue bar charts represent the map sizes of 20×20, 40×40 and 80×80 respectively. The results show that the optimized algorithm's path length is not shorter compared to the traditional A* algorithm and the JPS algorithm, but the overall performance is slightly better.

5. Conclusions
The search efficiency of the traditional A* algorithm is low, which results in a large number of redundant points, especially in the large map. When the number of obstacles increases, the safety of the path cannot be guaranteed. In this paper, an optimized A* algorithm is proposed to improve the heuristic function.
by adaptive dynamic weighting, which improves the search efficiency of the algorithm. The JPS algorithm reduces a lot of redundant points and memory expenses. The dynamic stretching of the obstacle makes the track maintain a safe distance from the obstacle and reduces the possibility of collision. Compared with the traditional A* algorithm and JPS algorithm, the simulation experiment shows that the improved A* algorithm has shorter running time, lower risk coefficient, fewer evaluation nodes and better comprehensive performance. In the near future, the defects of the algorithm will gradually improve. On the basis of the combination of software and hardware, such as multi-sensor fusion and machine learning algorithm, mobile robots can solve the problem of path planning faster, and make the development of mobile robots and even society leap forward.

Acknowledgments
This work is Supported by the Open Research Fund of AnHui Key Laboratory of Detection Technology and Energy Saving Devices, AnHui Polytechnic University (No. DTESD2020A06), 2021 Anhui University Graduate scientific research project (YJS20210447), Wuhu City Science and Technology Plan Project (2021cg21) and National College Student Innovation and Entrepreneurship Training Program (No. 202010363039).

References
[1] Dang H S, Lu X R, Sun J L, et al. Design of Vision-Based Drug-Grabbing Robot System[J]. Laser & Optoelectronics Progress, 2021, 58(2): 0210009.
[2] Chang L, Lee S, Varnhagen S, et al. Path planning for autonomous vehicles using model predictive control[C]//Intelligent Vehicles Symposium (IV), June 11-14, 2017, Los Angeles, CA, USA. New York: IEEE, 2017: 174-179.
[3] Song Q, Zhao Q L, Wang S X, et al. Dynamic Path Planning for Unmanned Vehicles Based on Fuzzy Logic and Improved Ant Colony Optimization[J]. IEEE Access, 2020, 8: 62107-62115.
[4] Sung I, Choi B, Nielsen P. On the training of a neural network for online path planning with offline path planning algorithms[J]. International Journal of Information Management, 2021, 57: 102142.
[5] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science & Cybernetics, 1968, 4(2): 100-107.
[6] Sang H, You Y, Sun X, et al. The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations[J]. Ocean Engineering, 2021, 223: 108709.
[7] Zhou H. Research on Motion Path Planning of Industrial Robot Based on Genetic Algorithms[C]//2019 International Conference on Artificial Intelligence and Advanced Manufacturing, October 16-18, 2019, Dublin, Ireland. New York: IEEE, 2019: 578-582.
[8] Botea A, Müller M, Schaeffler J. Near optimal hierarchical path-finding (HPA*)[J]. Journal of Game Development, 2004, 1(1): 7-28.
[9] Ren Y Y, Song X R and Gao S. Research on Path Planning of Mobile Robot Based on Improved A* in Special Environment[C]// International Symposium on Autonomous Systems (ISAS), May 29-31, 2019, Shanghai, China. New York: IEEE, 2019: 12-16.
[10] Ju C Y, Luo Q H, Yan X Z. Path Planning Using an Improved A-star Algorithm[C]//2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), October 23-25, 2020, Jinan, China. New York: IEEE, 2020: 23-26.
[11] Zheng X, Tu X W, Yang Q H. Improved JPS Algorithm Using New Jump Point for Path Planning of Mobile Robot[C]//2019 IEEE International Conference on Mechatronics and Automation(ICMA), August 4-7, 2019, Tianjin, China. New York: IEEE, 2019: 2463-2468.
[12] Zhang X Y, Duan H B, Luo Q N. Levenberg-Marquardt based artificial physics method for mobile robot oscillation alleviation[J]. Science China Physics, Mechanics & Astronomy, 2014, 57(9): 1771-1777.