

An Improved Hybrid A* Algorithm of Path Planning for Hotel Service Robot

Xiaobing Cao¹, Yicen Xu², Yonghong Yao³, Chenbo Zhi⁴

School of Control Technology, Wuxi Institute of Technology, Jiangsu Wuxi, 214121, China^{1,3}

School of Intelligent Equipment and Automotive Engineering, Wuxi Institute of Commerce, Jiangsu Wuxi, 214153, China²
Xi'an Research Institute of Huawei Technology Co, Ltd, Shaanxi Xi'an, 710075, China⁴

Abstract—Due to the increasing demand for unmanned in the hotel industry in recent years, how to efficiently use hotel service robots to further improve the efficiency of the hotel industry has become a hot research issue. To solve the problems of lengthy path-finding time and poor route security in conventional service robots in complex environments, the current study provides an improved A* path-finding algorithm for application in the hotel environment. Firstly, the conventional A* algorithm is combined with bidirectional search and Jump Point Search (JSP) algorithm, which makes the search more effective. Secondly, the traditional A* algorithm is combined with the security weight square matrix to make the path trajectory safer. A cubic spline interpolation is chosen to smoothen the transitions at the corners planned by the improved A* algorithm. Simulation experiments were done on grid maps with 10*10, 20*20 and 50*50 sizes. Compared with the conventional A* algorithm, the search time were decreased by 67%, 77% and 95% respectively. The number of search nodes was decreased by 80%, 76% and 95%, respectively. Meanwhile the distance between the robot and the obstacles was increased. The results indicate that the improved A* algorithm suggested in the present research can ensure the path trajectory safer while keeping the path search efficiency higher.

Keywords—Path planning; bidirectional A* algorithm; JPS algorithm; security weight square matrix; cubic spline interpolation method

I. INTRODUCTION

Path planning is an important technology to ensure the hotel service robot can complete the corresponding tasks; the purpose is to plan a reasonable and safe route from the starting point to the target point without collision based on the corresponding task requirements [1]. For the hotel scene with high real-time requirements and a complex environment, it is necessary to design an algorithm to consider both real-time and security and increase the operation efficiency of the service robot. This research aims to investigate the A* algorithm [2] for a fast and secure path planning problem.

Many scholars have done various investigations on the path planning of mobile robots. Regarding various search environment information, the planning is classified into global and local path planning. Among them, A* algorithm [3], Dijkstra algorithm [4], RRT* algorithm [5] are relatively common global path planning algorithms, while local path planning algorithms include dynamic window method, artificial potential field method and genetic algorithm [6], [7]. Although several researchers have researched this subject,

many deficiencies still exist. For example, in [8], a novel fusion algorithm of jump-A* algorithm and Dynamic Window Approach (DWA) was presented to satisfy global optimality and path smoothness performance requirements in robot path planning. However, the security of the planning path is unsatisfactory. To make robot movements safe in a realistic environment, Zhang et al. [9] considered node-to-obstacle distances in the A* extension node and used the threat generation value to assess node cost. That algorithm guarantees the safety of the path, and its computation time is also increased. Some three-path smoothers have been designed by Song et al. [10] for optimizing the paths, but the algorithm's time complexity is sensitive to the scale of the node number. In [11], a Bi-directional Adaptive Probabilistic Method having a Triangular Segmented Interpolation was developed to increase the real-time performance of path search by an Adaptive-RRT approach and a Bi-directional scheme. However, there are still too many redundant nodes. We synthesize the solutions involved in the above literatures, and organize their methods, advantages and disadvantages in the Table I.

TABLE I. ADVANTAGES AND DISADVANTAGES OF EXISTING METHODS

Method	Advantages (well considered)	Disadvantages (not fully considered)
fusion of jump-A* algorithm and DWA	computation time and path smoothness	security
A* algorithm taking the safety cost into account	safety and smoothness	computation time
A* algorithm with three path smoothers	Path smoothness	computation time
Bi-directional adaptive probabilistic method	computation time and path smoothness	redundant nodes

As shown in Table I, the related research work of other scholars on the path planning algorithm have been comprehensively studied and improved from the aspects of real-time, security and smoothness, and satisfactory results have been achieved. Thus, the research in this paper adopts a new improved way to achieve better performance of A* algorithm in the hotel environment. To realize that the hotel service robot can travel in the globally optimal path and keep a safer distance from obstructions, this paper proposes a new solution. A global path planning algorithm combined with bidirectional search and Jump Point Search (JSP) algorithm, also combined with the security weight square matrix is suggested in the present study in terms of improved A*.

The contributions of the improved hybrid algorithm recommended in the present research to the path planning

problem are as follows. 1) The concept of bidirectional search [12] and JPS jump point search [13], [14] is used to promote the searching efficacy of the traditional A* algorithm. 2) The security weight square matrix with the idea of multi-neighborhood search [15], [16] is suggested as a new method for helping the robots prevent temporary obstacles in the global path. According to the requirements of different safety distances, that manner sets the corresponding neighbourhood matrix to complete the detection of obstacles, making the path trajectory safer. 3) Cubic spline interpolation is employed to improve the smoothness of the path [17], which is unsmooth at the corner.

The remaining of the paper is as follows. Section II first reviews the traditional A* algorithm, then discusses the bidirectional A* algorithm regarding Jump Point Search and presents a cubic spline interpolation method to smooth the turning of the path. Thirdly, it introduces the security weight square matrix. Finally, the steps of the improved hybrid A* algorithm proposed in the present article are summed. Section III represents the experimental results of the simulation. Section IV concludes the whole research in summary and provides future research directions.

II. IMPROVED A* ALGORITHM

A. Traditional A* algorithm

A* algorithm [18], [19] is a heuristic search algorithm relying on the cost function. It can be considered as the most efficient direct search method to solve the shortest path in static road networks, usually used in global planning grid map scenes. The algorithm can evaluate the cost value of nodes around the existent path and find the node with the lowest cost value as the next moving point, and so on, until the robot reaches the target point. The formula of the cost function is defined in Equation (1):

$$F(n) = G(n) + H(n) \quad (1)$$

Where $F(n)$ is the cost value of the current moving node, $G(n)$ represents the cost value from the starting node to the current moving node, and $H(n)$ represents the cost value from the current moving node to the target node.

B. Real-Time Improvement of A* Algorithm

The traditional A* algorithm searches from the starting point to the target point, but its search effectiveness is not optimal. Hence, an improved A* algorithm is suggested by some scholars [20]–[23] that changes the one-way search of the primary algorithm to the bi-directional search. The improved A* algorithm searches from the starting point and the target point simultaneously, which greatly enhances the search effectiveness. The specific formula is as follows.

$$\begin{cases} F_1(n) = G_1(n) + H_1(n) \\ F_2(n) = G_2(n) + H_2(n) \end{cases} \quad (2)$$

Where $F_1(n)$ is the cost value from the starting point to the target point and $F_2(n)$ is the cost value from the target point to the starting point. When one party detects the checked node of the other party, the value of $F_1(n) + F_2(n)$ is the minimum and the search ends.

For the bidirectional search, it is easy to form a parallel search range while the obstacles exist between the starting and target points, increasing traversal nodes. Hence, this work mixes the JPS algorithm with the A* algorithm, greatly reducing traversal nodes by selecting corresponding hop points. OpenList1 and OpenList2 are established in this paper, respectively, representing the search from the starting point to the target point and vice versa. When searching the path, the two search directions are carried out alternately. The jump point is added to Openlist1 when searching in the forward direction. Then, the forward search is stopped and changed to the reverse direction. When searching in the reverse direction, the jump point is added to Openlist2, then stop the reverse direction search and start the forward direction search again. Carry out the forward and reverse search alternately until the same jump point is added to Openlist1 and Openlist2, which means the optimum path is found.

The improved A* algorithm can search a suitable path quickly, but a large turning point still exists at the corner. A relatively smooth turning is usually required for the specific complex hotel scenario. The present paper selects the cubic spline interpolation [24], [25] method to smoothen the turning trajectory in the improved A* algorithm. Its detailed definition is shown as follows.

For a piecewise function $S(x)$, if $S(x) = S_i(x)$ is a cubic polynomial in every interval $[x_i, x_{i+1}]$ ($i = 0, 1, 2, 3, \dots, n - 1$) composed of $n + 1$ discrete points and $S(x_i) = y_i$ ($i = 0, 1, 2, 3, \dots, n$) is satisfied at each point, this function $S(x)$ is a cubic spline interpolation function. Since $S(x)$ is a second derivative and the second derivative is continuous in the continuous interval $[m, n]$, the $S(x)$ curve can be understood as a smooth curve, so the path it fits at the corner is naturally smooth.

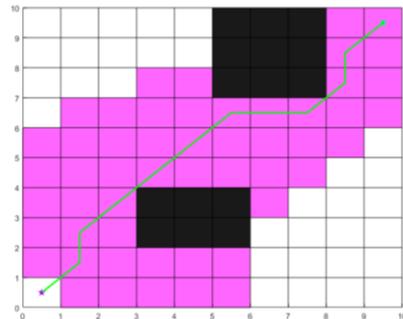


Fig. 1. Traditional A* algorithm.

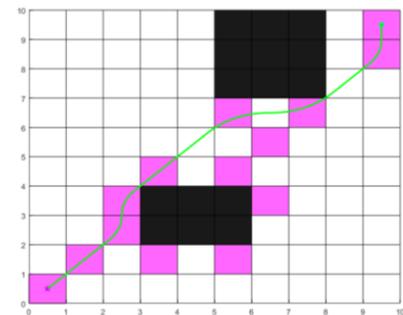


Fig. 2. A* algorithm integrated with JPS and bidirectional search.

Fig. 1 and Fig. 2 show the simulation experiments of the traditional A* algorithm and the A* algorithm integrating JPS and bidirectional search, where the black square shows the obstacle, the pink square represents the search node, and the green line refers to the planned path. The simulation experiments above demonstrate that the improved algorithm has fewer search nodes and better real-time performance. Although it smooths the turning of the path, a big risk of collision between the hotel service robot and obstacles still exists because of the close distance. Therefore, how to promote security is a crucial problem.

C. Security Improvement of A* Algorithm

In a traditional A* algorithm, the path planning situation shown in Fig. 3 usually occurs where the green S square denotes the starting point, the red G square denotes the target point, the black square denotes the obstacle, and the blue line denotes the planned path. However, it ignores the volume of the robot in practical application, which easily causes a false collision at the position marked X. This paper introduces the idea of a security weight square matrix for ensuring that the mobile robot can effectively keep a safe distance from obstacles, to increase the security of the algorithm.

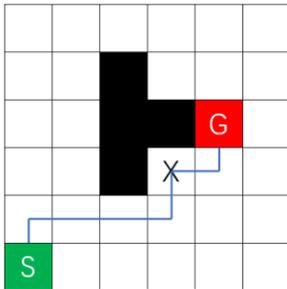


Fig. 3. Path planned via A* algorithm before security improvement.

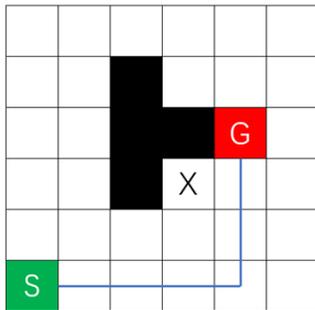


Fig. 4. Path planned via A* algorithm with security improvement.

According to the multi-neighbourhood search idea, the present work designs a corresponding neighbourhood matrix to complete the detection of obstacles according to the requirements of different safety distances. Because the map environment type in this paper is a grid map, the minimum safe distance from obstacles is set as the distance of a grid, and the size of the corresponding neighbourhood matrix is 3*3, which is also known as the basic comparison matrix. Taking the basic comparison square matrix as an example, the central element of the square matrix is set as the current path node with a value of 0 and other position values are set to 1, which can be shown in the following matrix.

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (3)$$

In the process of path search, if an obstacle is detected within the range of the matrix, the element value of the corresponding position should be changed from 1 to 2 to generate a security weight square matrix, indicating that there is an obstacle. For example, for the path node X in Fig. 3, its security weight square matrix is.

$$X_B = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (4)$$

Set d as the safety factor. If d is larger, the degree of path safety is higher. If there exists 2 in the square matrix, set d to 5 (the actual situation can be selected on its own), indicating obstacles near the current node. Otherwise, d is 0. The detailed expression can be written as follows.

$$d = \begin{cases} 5 & \exists X_{B_{ij}} = 2 \\ 0 & \forall X_{B_{ij}} \neq 2 \end{cases} \quad (5)$$

The functional cost of the improved A* algorithm is shown in formula (6), where m is the number of 2 in the statistical basis comparison matrix, also known as the obstacle influence factor. The larger its value is, the more obstacles near the current node. As shown in Fig. 4, due to the increase in the cost value for the original path node, the path node is re-selected to keep the path away from obstacles.

$$f(n) = g(n) + h(n) + md \quad (6)$$

The matrix transformation can be based on the basic comparison matrix to set other safety distances. Using formula (8) with the transformation matrix A, the comparison matrix C of $k \times k$ (k is odd) with the central element value of 0 and the remaining elements value of 1 is obtained.

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ \dots & & \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}_{k \times 3} \quad (7)$$

$$\begin{cases} C = \frac{1}{8}ABA^T \\ C_{\frac{k}{2}+1, \frac{k}{2}+1} = 0 \end{cases} \quad (8)$$

According to formula (4), the comparison square matrix C generates the security weight square matrix and the path node is re-selected. The flowchart of security improvement for the A* algorithm is shown in Fig. 5.

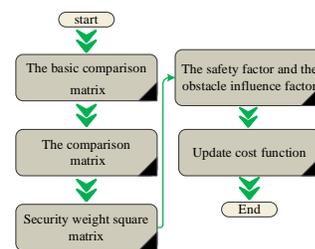


Fig. 5. A* algorithm security improvement flow chart.

D. Improvement of Hybrid A* Algorithm

To effectively promote the path planning algorithm in a specific service robot in the hotel environment, the present study mixes a bi-directional A* algorithm with a JPS algorithm to enhance its real-time performance. Simultaneously, it combines the idea of a security weight square matrix to enhance the algorithm's security. Finally, the cubic spline interpolation method is applied to smooth the improved A* algorithm at the path corner to construct the improved A* global path planning algorithm. The flow chart of the improved A* algorithm is represented in Fig. 6. The detailed steps include:

1) Create a grid map, initialize the corresponding parameters, starting point position and target point position, establish OpenList1 and OpenList2, select the first hop point and add it to OpenList1.

2) Set the safety distance. This paper uniformly sets the safety distance to facilitate comparison as 1 grid distance.

3) Compare the distance between the hop point to be selected and the obstacle, and add it to OpenList1 if it meets the conditions during the forward search.

4) Judge whether the jump point exists in OpenList2. If so, smooth the turning of the path. If not, stop the forward search and change to the reverse search.

5) Compare the distance between the jump point to be selected and the obstacle, and add it to OpenList2 if it meets the conditions during the reverse search.

6) Judge whether the jump point exists in OpenList1. If so, smooth the turning of the path. If not, stop the reverse search and change to the forward search.

7) Loop steps 3 to 6.

8) Output the optimal path.

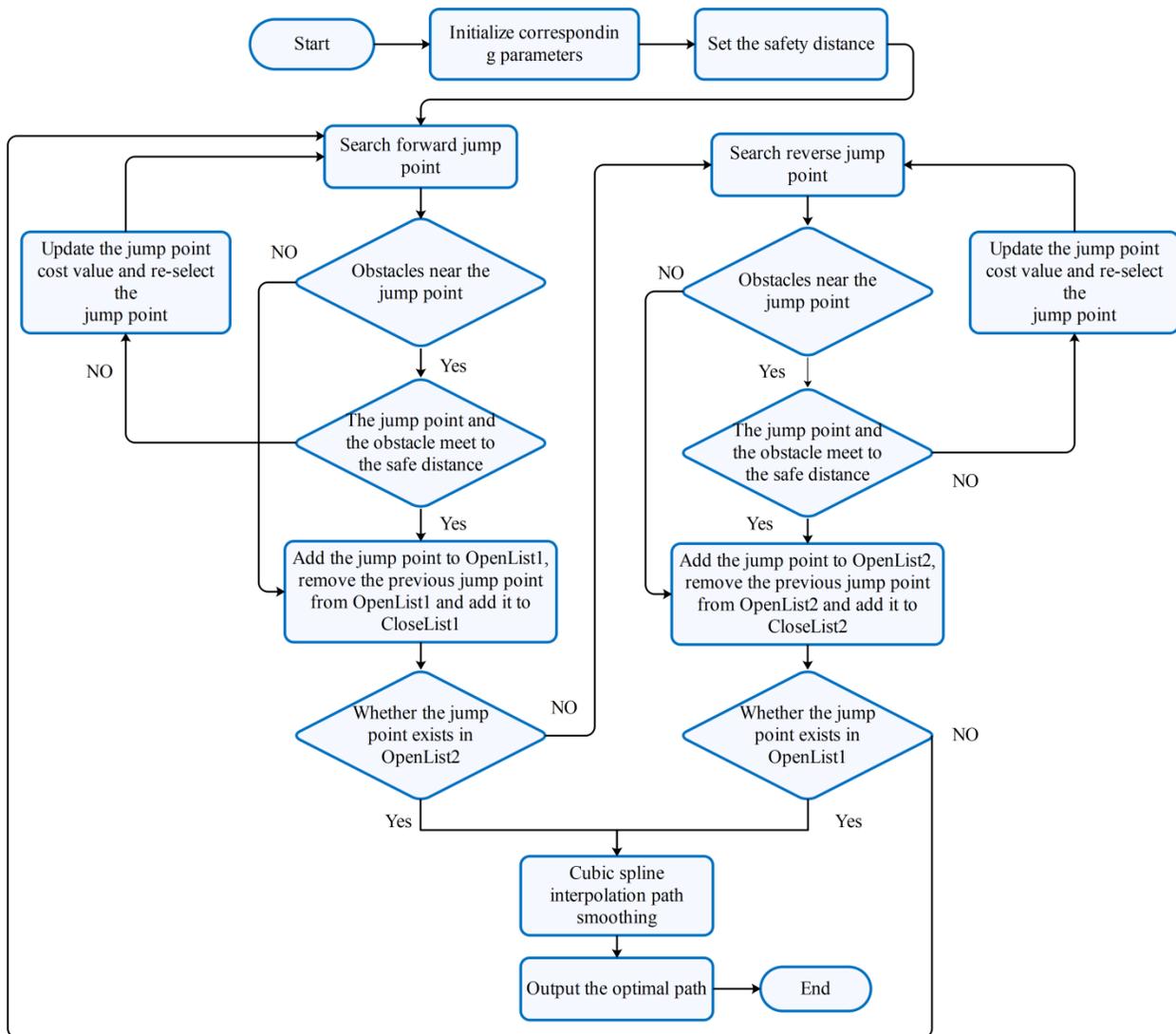


Fig. 6. Flow chart of improved A* algorithm.

III. SIMULATION EXPERIMENTAL ANALYSIS

A. Simulation Scenarios and Conditions

Simulation experiments are done in this part to validate the improved A* algorithm in mobile robots' authentic path-planning process. The computer parameters of the simulation environment are system Windows 10, CPU Intel Core i5 4210h, memory 8GB, compilation environment MATLAB 2016b. Since the research work of this paper comes from a real hotel robot project, we simulate three sets of grid map environments with different sizes and several sparse obstacles. These grid maps are associated with a real hotel scene and the sizes of them are $10 * 10$, $20 * 20$, $50 * 50$ grids, respectively.

B. Results and Discussion

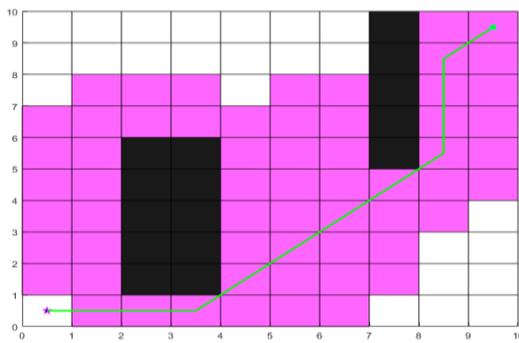
The simulation results have been presented in Fig. 7 to 9, where the black square shows the obstacle, the pink square indicates the search node, and the green line refers to the planned path. The test of each group presents an analytical contrast between the results of these two methods based on the nodes and path length.

As shown in Fig. 7, the improved hybrid A* algorithm reduces the number of nodes from 61 to 12 compared with the traditional algorithm. In the middle size map as shown in Fig. 8, the improved hybrid A* algorithm reduces the number of nodes from 139 to 33, and in the large map as shown in Fig. 8, the number of nodes is reduced from 787 to 42. It can

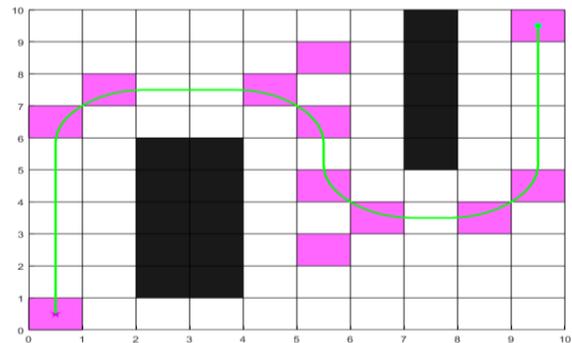
be seen from the comparison of Fig. 7(a) and Fig. 7(b), the safe distance from the obstacle is increased about 1 grid and the path is smoother in the simulation results for the improved hybrid A* algorithm, and the same results can also be obtained from Fig. 8 and Fig. 9.

Total counts of nodes, path lengths and search time are further used to evaluate the improved algorithm's efficacy, as shown in Table II. It can be seen that compared with the traditional algorithm, the search time of the improved A* algorithm for $10 * 10$, $20 * 20$, $50 * 50$ environmental raster maps are reduced from 0.082 s to 0.027 s, from 0.198 s to 0.046 s, and from 1.211 s to 0.065 s, respectively. Although the path length increased, the search time and the number of nodes have decreased. By comparing the results shown in Table I, it can also be seen that the improved A* algorithm has obvious priority regarding search time and node number with the expansion of the map area. Although the path planning length is relatively long, it can guarantee the safety of the path trajectory.

Consequently, from the previous analysis of the results, we can conclude that the improved A* algorithm suggested has fewer search nodes and less search time while keeping safe distance compared with traditional A* algorithm. It is proved that the algorithm presented in the paper enables the search for an optimum path under the condition of ensuring safety and shows good robustness and real-time performance.

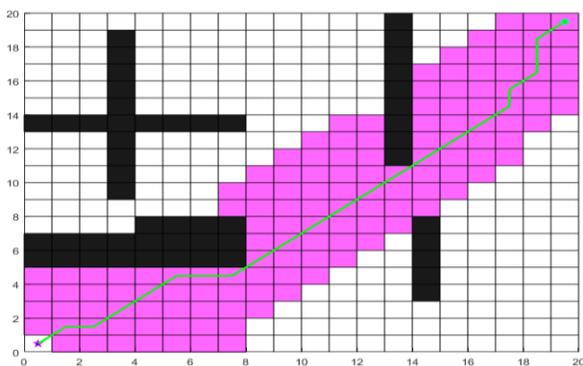


(a) Simulation results for the traditional A* algorithm

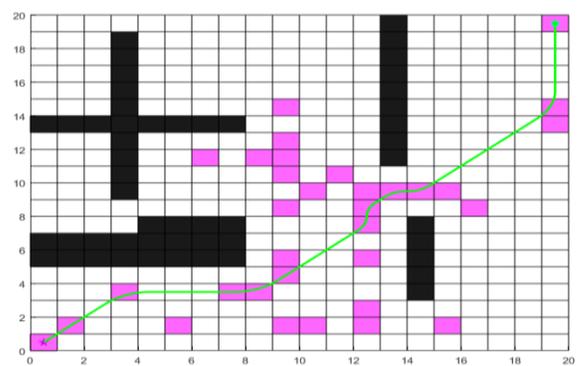


(b) Simulation results for the improved hybrid A* algorithm

Fig. 7. Path planning in a $10 * 10$ environment grid map by using different algorithms.

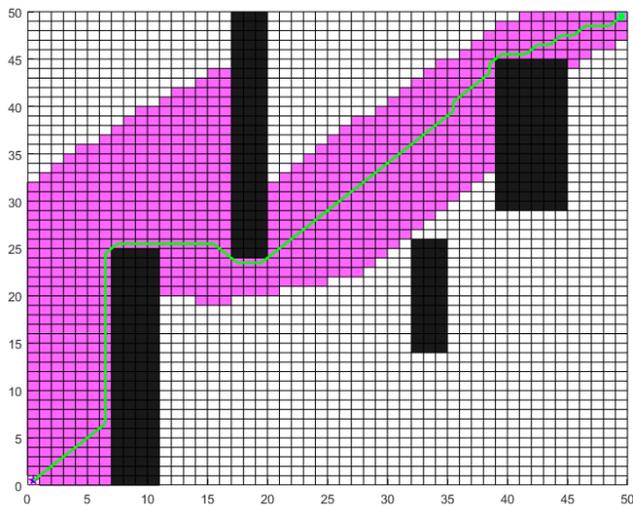


(a) Simulation results for the traditional A* algorithm

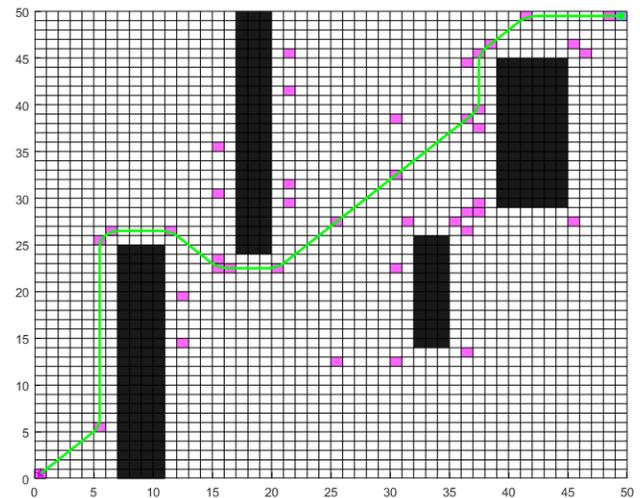


(b) Simulation results for the improved hybrid A* algorithm

Fig. 8. Path planning in a $20 * 20$ environment grid map by using different algorithms.



(a) Simulation results for the traditional A* algorithm



(b) Simulation results for the improved hybrid A* algorithm

Fig. 9. Path planning in a 50*50 environment grid map by using a different algorithm.

TABLE II. DATA COMPARISON BETWEEN TRADITIONAL A* ALGORITHM AND IMPROVED HYBRID A* ALGORITHM

Map size	Search time /s		Number of nodes		Path length	
	Traditional	Improved	Traditional	Improved	Traditional	Improved
10*10	0.082	0.027	61	12	14.5	23.1
20*20	0.198	0.046	139	33	28.6	29.9
50*50	1.211	0.065	787	42	82.7	87.0

IV. CONCLUSION

In the complex regional environment of the hotel, even though the traditional A* algorithm is proper to plan a reasonable shortest path, several inflection points are available in the path. That makes the hotel service robot too close to the obstacle during driving and will seriously influence the safe movement of mobile robots. Numerous nodes are evaluated in the process of path planning simultaneously, which makes its real-time performance not good enough. To overcome such problems, the current paper suggests an improved A* algorithm. First, the conventional A* algorithm combines bidirectional search and JPS hop point selection to enhance search efficiency. Then, the traditional A* algorithm is combined with the security weight square matrix to make the path trajectory safer. Finally, cubic spline interpolation is employed to smoothen the turning path trajectory. The experimental results show that the improved algorithm is productive and better than the conventional A* algorithm in the number of search path nodes, path security and path smoothness.

The limits of the proposed algorithm is that it is only applicable to static environment, and does not have the ability to avoid dynamic obstacles. Since there are still dynamic areas in the hotel environment, such as pedestrians and other dynamic obstacles, we will study how to embed dynamic path planning into this algorithm to meet the specific needs of the hotel mission in future studies.

COMPETING OF INTERESTS

The authors declare no competing of interests.

AUTHORSHIP CONTRIBUTION STATEMENT

Xiaobing Cao: Writing-Original draft preparation
Conceptualization, Supervision, Project administration.
Yicen Xu: Language review, Methodology, Software
Yonghong Yao: Formal analysis
Chenbo Zhi: Validation

AVAILABILITY OF DATA AND MATERIALS

On Request

DECLARATIONS

not applicable

REFERENCES

- [1] B. K. Patle, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.
- [2] X. Sa, W. Huaiyu, and C. Zhihuan, "Research of mobile robot path planning based on improved A* algorithm," in *2020 Chinese Automation Congress (CAC)*, IEEE, 2020, pp. 7619–7623.
- [3] C. Cheng, X. Hao, J. Li, Z. Zhang, and G. Sun, "Global dynamic path planning based on fusion of improved A* algorithm and dynamic window approach," *Journal of xi'an jiaotong university*, vol. 51, no. 11, pp. 137–143, 2017.
- [4] S. Alshammrei, S. Boubaker, L. Kolsi, "Improved Dijkstra Algorithm for Mobile Robot Path Planning and Obstacle Avoidance" *Computers Materials & Continua*, vol. 72 no. 3, pp. 5939–5954, 2022.
- [5] P. Xin, X. Wang, X. Liu, Y. Wang, Z. Zhai, and X. Ma, "Improved bidirectional RRT* algorithm for robot path planning," *Sensors*, vol. 23, no. 2, p. 1041, 2023.

- [6] K. Zhou, L. Yu, Z. Long, and S. Mo, "Local path planning of driverless car navigation based on jump point search method under urban environment," *Future Internet*, vol. 9, no. 3, p. 51, 2017.
- [7] M. Alireza, D. Vincent, and W. Tony, "Experimental study of path planning problem using EMCOA for a holonomic mobile robot," *Journal of Systems Engineering and Electronics*, vol. 32, no. 6, pp. 1450–1462, 2021.
- [8] L. Liu et al., "Global dynamic path planning fusion algorithm combining jump-A* algorithm and dynamic window approach," *IEEE Access*, vol. 9, pp. 19632–19638, 2021.
- [9] H.-M. Zhang, M.-L. Li, and L. Yang, "Safe path planning of mobile robot based on improved A* algorithm in complex terrains," *Algorithms*, vol. 11, no. 4, p. 44, 2018.
- [10] R. Song, Y. Liu, and R. Bucknall, "Smoothed A* algorithm for practical unmanned surface vehicle path planning," *Applied Ocean Research*, vol. 83, pp. 9–20, 2019.
- [11] S. Al-Ansary, S. Al-Darraj, A. Shareef, D. G. Honi, and F. Fallucchi, "Bi-directional Adaptive Probabilistic Method with a Triangular Segmented Interpolation for Robot Path Planning in Complex Dynamic-Environments," *IEEE Access*, 2023.
- [12] C. Li, X. Huang, J. Ding, K. Song, and S. Lu, "Global path planning based on a bidirectional alternating search A* algorithm for mobile robots," *Comput Ind Eng*, vol. 168, p. 108123, 2022.
- [13] K. Shi, Z. Wu, B. Jiang, and H. R. Karimi, "Dynamic path planning of mobile robot based on improved simulated annealing algorithm," *J Franklin Inst*, vol. 360, no. 6, pp. 4378–4398, 2023.
- [14] A. Janis and A. Bade, "Integration of Enhanced Jump Point Search Algorithm and Modified Bresenham Technique for Path Finding in Grid-Based Map Environment," *Adv Sci Lett*, vol. 24, no. 3, pp. 1582–1586, 2018.
- [15] X. Li, F. Liu, J. Liu, and S. Liang, "Obstacle avoidance for mobile robot based on improved dynamic window approach," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 25, no. 2, pp. 666–676, 2017.
- [16] F. P. Vista, A. M. Singh, D.-J. Lee, and K. T. Chong, "Design convergent Dynamic Window Approach for quadrotor navigation," *International journal of precision engineering and manufacturing*, vol. 15, pp. 2177–2184, 2014.
- [17] Z. Yang, J. Li, L. Yang, and H. Chen, "A Smooth Jump Point Search Algorithm for Mobile Robots Path Planning Based on a Two-Dimensional Grid Model," *Journal of Robotics*, vol. 2022, 2022.
- [18] Y. Li, J. Zhao, Z. Chen, G. Xiong, and S. Liu, "A robot path planning method based on improved genetic algorithm and improved dynamic window approach," *Sustainability*, vol. 15, no. 5, p. 4656, 2023.
- [19] F. Duchoň et al., "Path planning with modified a star algorithm for a mobile robot," *Procedia Eng*, vol. 96, pp. 59–69, 2014.
- [20] T. Wang, L. Zhao, Y. Jia, and J. Wang, "Robot path planning based on improved ant colony algorithm," in *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, IEEE, 2018, pp. 70–76.
- [21] H. Wang, X. Qi, S. Lou, J. Jing, H. He, and W. Liu, "An Efficient and Robust Improved A* Algorithm for Path Planning," *Symmetry (Basel)*, vol. 13, no. 11, p. 2213, 2021.
- [22] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment," *IEEE access*, vol. 9, pp. 59196–59210, 2021.
- [23] T.-W. Kang, J.-G. Kang, and J.-W. Jung, "A Bidirectional Interpolation Method for Post-Processing in Sampling-Based Robot Path Planning," *Sensors*, vol. 21, no. 21, p. 7425, 2021.
- [24] Y. Bai and H. Zhuang, "On the comparison of bilinear, cubic spline, and fuzzy interpolation techniques for robotic position measurements," *IEEE Trans Instrum Meas*, vol. 54, no. 6, pp. 2281–2288, 2005.
- [25] S.-H. Hong, L. Wang, T.-K. Truong, T.-C. Lin, and L.-J. Wang, "Novel approaches to the parametric cubic-spline interpolation," *IEEE transactions on Image Processing*, vol. 22, no. 3, pp. 1233–1241, 2012.