Towards Path Planning Algorithm Combining with A-Star Algorithm and Dynamic Window Approach Algorithm

Kaiyu Li[1], Xiugang Gong[2,*], Muhammad Tahir[3], Tao Wang[4], and Rajesh Kumar[5]
School of Computer Science and Technology, Shandong University of Technology, Zibo, Shandong, 255000, P.R. China[1, 2, 4]
Department of Computer Science, Mohammad Ali Jinnah University, P.E.C.H.S, Karachi, Sindh, 75400, Pakistan[3]
Department of Computer Science, University of Palermo, Italy[5]

***Abstract*—In the Automated Guided Vehicle (AGV) warehouse automatic guided vehicle system, the path planning algorithm for intelligent logistics vehicles is a key factor to ensure the stable and efficient operation of the system. However, the existing planning algorithms have problems such as single designing a route and the inability to intelligently evade moving barriers. The academic community has proposed various solutions to these problems, although they have improved the efficiency and quality of path planning to some extent, they have not completely solved problems such as poor safety in planning, the high number of path inflection points, poor path smoothness, easily getting stuck in deadlocks, and have not fully considered the running cost and practical implementation difficulty of algorithms. To address these issues, the article deeply researched traditional A* scheme and Dynamic Window Approach (DWA) technology and proposed designing a route method according to the fusion of the A* algorithm and DWA technology. The algorithm improved the A algorithm by introducing a sub-node optimization algorithm to solve problems for instance poor global path planning safety and easy deadlock. Moreover, the algorithm reduced the amount of global route reversal locations and increased path consistency by improving the evaluation function and removing redundant points of the A algorithm. Finally, by integrating the DWA algorithm, the intelligent logistics vehicle achieved dynamic obstacle avoidance capabilities for moving objects in the real world. Our simulations-based results on MATLAB framework show that the algorithm significantly improves path smoothness, path length, path planning time, and environmental adaptability compared to traditional algorithms, and basically meets the path planning requirements of the AGV system for intelligent logistics vehicles.**

Keywords*—AGV; path planning; A* algorithm; dynamic window approach

I. Introduction

In recent years, with the major breakthroughs in the P.R. China-Europe Railway Express, the China-Japan-Korea Free Trade Zone, and the Belt and Road Initiative (BRI) in Pakistan, China's flagship economic corridor economy has ushered in new development, and at the same time stimulated the rapid growth of the logistics industry. According to statistics from relevant departments, the total volume of China's logistics industry in 2022 will reach 347.6 trillion, a year-on-year increase of 3.4% [1]. At the same time, due to the continuous development of Artificial Intelligence (AI), warehouse-automated guided vehicles (automated guided vehicles, AGV) are commonly utilized in warehousing and distribution, logistics distribution, and other industries [2]. But at this stage, the AGV control system has a single path planning and fixed-point pickup, and cannot dynamically avoid dynamic obstacles [3]. The dynamic path planning of the AGV control system needs further research.

At present, many experts and scholars put forward different solutions. J Borenstein's team proposes a virtual stand that considers the dynamic behavior of fast-moving robots and solves the local minimum trap problem. Hao W's team used neural networks for path planning [4]. Chang et al. presented the SPEA2 method into the genetic algorithm to distribute the fitness of the population individuals and realized the balance of path smoothness, path length, and path difficulty [5], Xiong Lijun et al. introduced the initial grid transfer rule and changed the information The method of updating elements, deleting redundant nodes, DWA algorithm [6] for the design for regional paths and other methods can improve the rate of integration of the ant colony algorithm, the smoothness of the planned path, and the safety and reliability. Performance indicators such as smoothness, safety, and reliability are improved compared with traditional algorithms [7]. The K Dan team proposed the A* algorithm for the first time [8]. Yang Guihua and others adopted the method of cleaning the Close list to optimize the total quantity of vertices in the route that the A* method designed [9], while Yang Mingliang and others tried to integrate the A* algorithm and The DWA algorithm into achieves the global optimality of the planned route via avoiding obstacles the objective is to arrive [10]. Although the above method has improved the efficiency and quality of path planning to a certain extent, it has not completely solved the problems of too many vertices in the path planning, is not smooth, and is easy to fall into a deadlock, and has not considered the cost of algorithm operation and the difficulty of the actual implementation. Focusing on the real-world issues of adaptive obstacle-resistant routing for stored robotics, this study suggests a route modeling technique utilizing DWA and the typical A* technique.

The main contribution of this research work is as follows:

1) Firstly, the problems of the A* algorithm such as long search time, readily prone to global optimal, traversing obstacle vertices, and insufficient path smoothness are improved; a

*Corresponding Author.

globally optimal path is planned, and then combined with DWA to analyze the dynamic obstacle part.

2) Secondly, we create an adjacent adaptive obstacle-avoiding route to arrive at the desired route.

3) Finally, we conduct three sets of comparative computer simulations to verify that the method which is suggested performs superior to the traditional algorithm and the existing improved algorithm in both stationary and dynamic settings where the obstacle movement direction is known and unknown superiority.

The paper is structured into several sections. Section I introduces the field of automated guided vehicles for guided vehicle systems. Section II provides an environment model construction. Section III presents the basic algorithm. Section IV leads to the improved safety and reliability of the improved A* algorithm. Section V focuses on simulation-based experimental results Finally, Section VI concludes this research work and stipulates future research directions in Section VII.

II. Environment Model Construction

As shown in Fig. 1, the grid map is used to construct the virtual working environment map of the robot. In this two-dimensional map, the black grids represent obstacles, respectively S1, S2, ..., Sn, and the white grids represent no obstacles, while Point N and Point T represent the beginning and ending points, respectively. The serial numbers of grids are 1, 2, 3... from bottom to top and from left to right. What needs to be paid is that a distinct identification and associated two-dimensional dimensions are assigned for every grid, and the conversion method is the following:

$$\begin{cases} x_i = (mod(i, MM) - 0.5) * \beta \\ y_i = (NN + 0.5 - rup(20 - i/NN)) * \beta \end{cases} \quad (1)$$

Among them, x_i and y_i are the coordinates of the i-th grid in the two-dimensional graph; β represents the unit length of the grid; mod represents rounding; MM and NN are the grids in the row direction and column direction of the grid respectively number; rup is a custom function, representing rounding up.

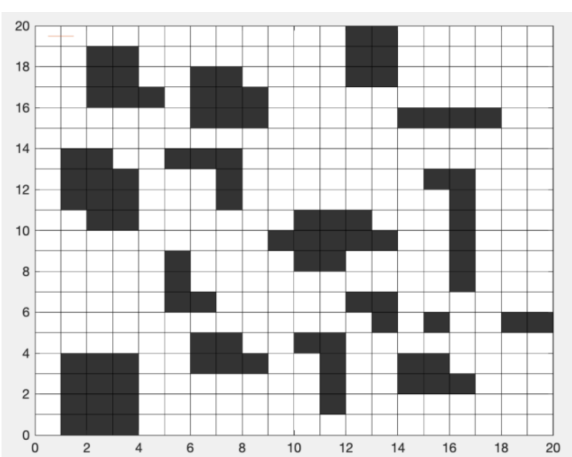

Fig. 1. Grid map of robot working environment.

III. Basic Algorithm

A. Traditional A* Algorithm

The shortest route among the two locations is determined by the most common methods are the Dijska algorithm [11] and A* algorithm. Compared with the Dijska algorithm, the A* method predicts the projected utility from the present location to the objective position in addition to recording the utility of getting to the beginning place. It has the advantages of faster speed and higher efficiency. It is a heuristic depth-first algorithm [12]. The assessing parameter of the conventional A* method is typical:

$$f(n) = g(n) + h(n) \ \# \quad (2)$$

Among them, f(n) depicts the car's assessment process in its present grid, g(n) indicates the actual utility value of the mobile robot from the initialing dot to the current dot, and h(n) denotes the heuristic performance [13], Euler distance was employed as the heuristic in the research [14].

$$H(n) = \sqrt{(N1_x - N2_x)^2 + \left(N1_y - N2_y\right)^2} \ \# \quad (3)$$

The A* technique's pathfinding strategy is displayed in the following Fig. 2:

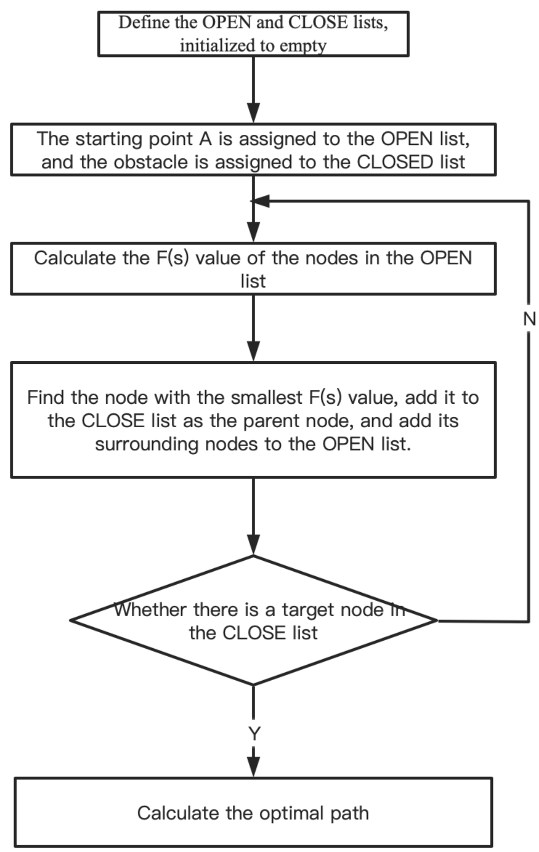

Fig. 2. A* Algorithm flow chart.

Step 1: Open the list and close the initial transmission of the list.

Step 2: Fill out the unfilled list with the initial location, then get the set of surrounding points and add it to the open list, and then the beginning spot should be taken off the list that is vacant and added to the closing condition of the closed list.

Step 3: Judging whether these surrounding point sets are already in the open list or not, then update the F and father points of these points and add them to the open list; if they are, then update G, F, and father points.

Among them, G stands for the utility associated with relocating initialing point A to a certain grid tile,

F represents the prediction function value of the point, and its value is equal to G+H, and H refers to the anticipated utility of moving from the specified square to the final result, that is, the utility of the heuristic function.

Step 4: then obtain the surrounding point set, find the point with the smallest F from the surrounding point set, then remove the point that finds the smallest F from the open list, and add it to the closed list.

Step 5: judge whether the opening list is empty, if it is empty, it means that the path does not exist, if it is not empty, execute Sep3 in a loop.

Step 6: If the objective grid is in the "open list", it means that the route has been identified, the algorithm ends, and the path planning is completed.

The classic A* algorithm has some issues, as the information mentioned shows:

1) The planned path is rough, and the inflection point is too close to static obstacles, so it cannot be directly applied to the movement of smart cars in real life.

2) It can only run in a static environment. When the working environment changes, it still plans the path according to the environmental grid map before modification, resulting in the existence of theory and failure of practice, and even causing safety hazards.

B. Dynamic Programming (DWA) Algorithm

To address the drawbacks of the confusing decision-making introduced by the A* technique, which only focuses on the path to the destination, and overlooks the impact of dynamic barriers on the performing route's track on the vehicle's route selection, the DWA (dynamic window technique) algorithm is a more suitable solution plan [15].

The fundamental idea is to collect different sets of rates in the velocity domain (v, w), and model the course of these velocities over a predetermined amount of time, and the rating function should be used to assess these motions, and decide on the corresponding optimal trajectory (v, w) Drive the robot to move[16].

There are three main steps in the operation of the algorithm, specifically:

1) Smart car kinematics model: As shown in Fig. 3, the smart car belongs to omnidirectional movement. In the robot coordinate system, there is the speed Vx in the X direction and the speed Vy in the Y direction [17]. The relationship between the position and speed at time t and time $t + \Delta t$ is as follows:

$$\begin{bmatrix} x(t+\Delta t) \\ y(t+\Delta t) \\ V_x(t+\Delta t) \\ V_y(t+\Delta t) \\ \theta(t+\Delta t) \\ \omega(t+\Delta t) \end{bmatrix} = \begin{bmatrix} x(t) + V_x(t) * \cos(\theta(t)) * \Delta t - V_y(t) * \sin(\theta(t)) * \Delta t \\ y(t) + V_x(t) * \sin(\theta(t)) * \Delta t - V_y(t) * \cos(\theta(t)) * \Delta t \\ V_x(t) + \alpha_x(t) * \Delta t \\ V_y(t) + \alpha_y(t) * \Delta t \\ \theta(t) + \omega(t) * \Delta t \\ \omega(t) + \alpha(t) * \Delta t \end{bmatrix} \# \quad (4)$$

It is especially pointed out that this model has two coordinate systems - the world coordinate system and the smart car coordinate system.

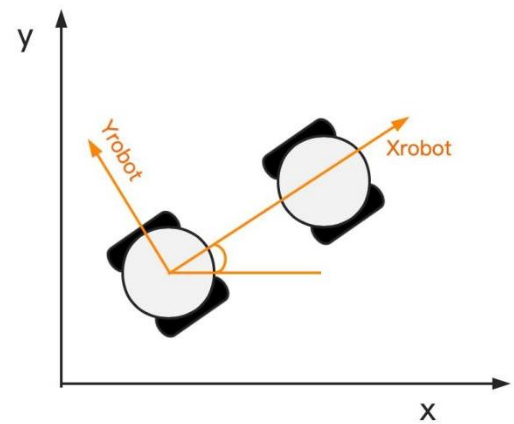

Fig. 3. Kinematics model of the smart car.

Vx(t) and Vy(t) in the above formula refer to the speed of the smart car in the x and y directions in the smart car coordinate system respectively.

2) Velocity space of the smart car: Theoretically speaking, the acceleration of the smart car should reach the maximum value that the machine motor can bear in an instant from the moment it stops to the start of acceleration, but in the actual environment, the car is limited by various factors, mainly including the following aspects:

a) The smart vehicle's velocity is its only constraint: Vs represents the range of vector speeds the vehicle is capable of which is affected by angular velocity and linear velocity:

$$V_s = \{(v,\omega) | v \in [v_{min}, v_{max}] \wedge \omega \in [\omega_{min}, \omega_{max}]\} \# \quad (5)$$

b) The moving trolley is affected by its own motor performance: In the actual operating environment, subject to cost and safety considerations, the acceleration of the car has a range limit, and the maximum acceleration and deceleration will take a certain amount of time to reach. The expressions are as follows:

$$V_d = \left\{(v,\omega) \middle| \begin{matrix} v \in [v_c - v_b * \Delta t, v_c + v_b * \Delta t\ ] \wedge \\ \omega \in [\omega_c - \omega_b * \Delta t, \omega_c - \omega_b * \Delta t] \end{matrix} \right\} \# \quad (6)$$

c) The mobile car is affected by obstacles: To make the car stop before it hits an obstacle and avoid damage to goods and property, the car should satisfy the following expression under the premise of maximum deceleration:

$$V_a = \left\{ (v,\omega) \middle| \begin{matrix} v \leq \sqrt{2dist(s,\omega)v_b} \wedge \\ \omega \leq \sqrt{2dist(s,\omega)\omega_b} \end{matrix} \right\} \# \qquad (7)$$

Among them, distance (s, w) shows the quickest path between the vehicle and the obstruction.

After the speed of the car passes through the three restrictions, the speed space will return to a certain extent, and it will change according to the changes in the linear velocity, angular velocity, and acceleration of the motor. This is the dynamic window of the car's movement [18], When the conditions are met, a sample of the speed space of the car, and the speed range in Fig. 4, below will be obtained:

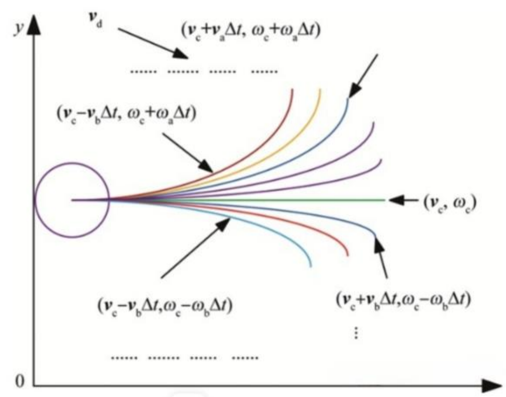

Fig. 4. Smart car speed range map [19].

3) Evaluation function: In general, the following is the evaluation process:

$$G(v,\omega) = \sigma \begin{pmatrix} \alpha * heading(v,\omega) + \\ \beta * dist(v,\omega) + \gamma * vel(v,\omega) \end{pmatrix} \# \quad (8)$$

Among them, heading (v, w) is the azimuth evaluation function: assess the angle between the intended location and the car's final route, given the present selected velocity; the main meaning of distance (v, w) is that the car is in the predicted. The final destination of the route is placed relative to the closest constraints on the map, and sample sites that are nearby the barrier are penalized to guarantee the car is able to prevent it and lessen the likelihood that it would collide with it; vel (v, w) is the current car, promoting the vehicle will help it swiftly reach its aim [20], α, β, γ are the weights, and σ is the smoothing coefficient, which is generally 1.

After obtaining a variety of trajectories through the above methods, the current optimal speed and the best trajectory are selected through the evaluation function, to drive the car to avoid obstacles as much as possible. But this method has the blindness and randomness of path selection.

IV. Improved A* Algorithm

A. Improve Safety and Reliability

Although the traditional A* scheme contains several benefits fast speed and global routing in path routing, it has the disadvantages of a large turning range, many trajectory polylines, and the initial path of turning to being close to obstacles. The following Fig. 5 shows the domain algorithm of the traditional A* scheme for routing:

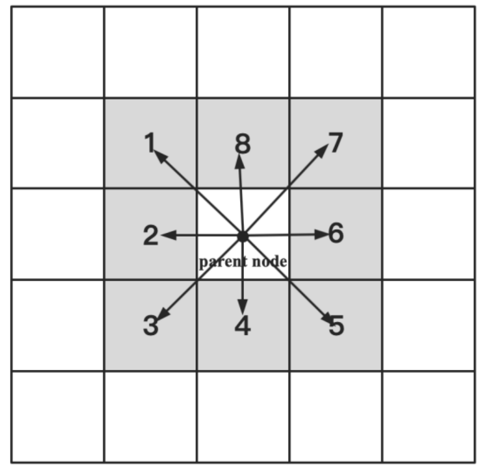

Fig. 5. Schematic diagram of the domain of the A* algorithm.

From the analysis in Fig. 5, we know that the reason for the insufficient uniformity of the planned route is that the type A* technology adopts one 3*3 domain algorithm, leading to the turning range of the smart car. This causes the issue that the planned car path is insufficient near static barriers.

To fix the challenge, the improved A* scheme needs to adopt a sub-node optimization algorithm in the route. Its core is to select an appropriate method to remove redundant nodes according to obstacles at different positions, to prevent the phenomenon in which the car traverses obliquely through the apex of obstacles and ensure that the car Safe distance from obstacles.

As shown in Table I, sub-nodes 1, 3, 5, and 7 are divided into X-type nodes, and sub-nodes 2, 4, 6, and 8 are divided into N-type nodes. According to the division method, the selection method of the sub-node optimization algorithm is:

TABLE I. Sub-Node Optimization Algorithm

Obstacle node type	***Handling measures***	***The number of remaining child nodes***
X	do not deal	8
N	removes two child nodes adjacent to the obstacle child node	6

B. Path Smoothness Optimization

After adopting the above-mentioned sub-node optimization algorithm, although the phenomenon of obliquely crossing the obstacle vertex can be avoided, it also strengthens another shortcoming of the A* algorithm - the problem that the path is not smooth and the path length is not optimal. In order to solve this problem, the improved A* algorithm adopts the method of twice smoothness optimization.

1) The first smoothness optimization: The first smoothness optimization is achieved by improving the evaluation function of the A* algorithm. By increasing the weight value of h(n) to increase the evaluation value G of the correct node, the algorithm selects a child node closer to the end position. The function expression is as follows:

$$f(n) = g(n) + \left(1 + \frac{r}{R}\right)h(n)\# \qquad (9)$$

Where *r* is the space between the beginning position and the goal point, where *R* is the value between the present and objective locations.

2) The second smoothness optimization: When the classical A* algorithm plans the path, there will be several redundant connections in enormous amounts, which will increase the length of the path and make the result not the optimal solution. Therefore, the A* algorithm for the second smoothness improvement adopts two methods to remove redundant nodes:

a) Remove collinear nodes, such as five nodes N_4, N_5, N_6, N_7, and N_8 belong to collinear nodes, then the child nodes of N3. The next step should be to change to the node with obstacles in the first domain of the collinear node, that is, N_8.

b) Eliminate the right-angle inflection point, when the next two steps of the parent node N_i are still in the domain of the parent node, it is determined that the path planning is in progress There is a right-angle inflection point. Currently, whether the child node of the domain union of the three parent nodes is an obstacle is judged. If not, the next hop of N_i is changed to N_i+2 instead of N_i+1. After two path smoothness optimizations, the path planning demonstration diagram of A* is shown in Fig. 6.

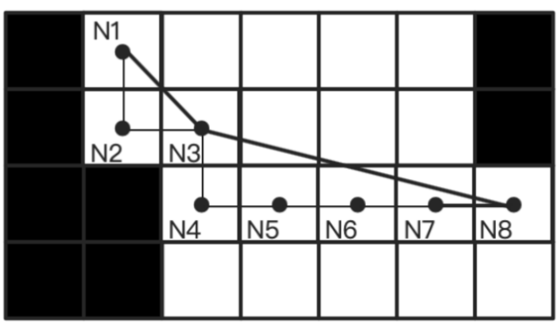

Fig. 6. A* path planning.

C. Integrated Obstacle Avoidance Strategy

Based on the above improvement scheme, the 9*9 improved A* algorithm incorporates the DWA algorithm, the A* algorithm is used for global path planning, a reference route is provided for DWA, and points are selected for local obstacle avoidance path planning, which solves the problem of the DWA algorithm. The blindness and randomness of the path selection realize the optimal combination of the shortest path and dynamic obstacle avoidance. At the same time, in order to ensure the path smoothness of the car at the turning point, and to ensure that the smart car can slow down in advance when encountering obstacles or turning, so as to ensure the safety of the goods and the optimization of the path length, the improved algorithm proposes a The new DWA evaluation algorithm based on the globally optimal path, its expression is as follows:

$$G(v,\omega) = \sigma(\alpha * heading(v,\omega) + \beta * dist(v,\omega) + \gamma * vel(v,\omega)) + \eta * Yheading(v,\omega,G_i)\ \# \qquad (10)$$

In the formula, α, β, γ, σ, and η are weighted, and σ is a smoothing coefficient, usually, its value is 1. Heading () is the azimuth evaluation function, distance () measures the separation between the course of motion and the obstruction, vel () is the evaluating process, Y heading () is the deviation function from the sub-target point to the end of the path, where Gi represents the current car The subgoal point closest to the global planning path. Therefore, this paper implements a global path planning algorithm that integrates obstacle avoidance, avoids the blindness and randomness of the classical DWA algorithm at the beginning of path selection, and effectively ensures the global route is followed in the best possible way by cutting the path's width and design time, Superiority.

V. Simulation-based Experimental Results

To assess how well the enhanced technique performs and verify the feasibility of the algorithm, the algorithm validity verification experiment and the algorithm comparison verification experiment were carried out. Both experiments were carried out on the Matlab R2023a simulation platform, the computer operating system version is MacOS 11, the processor is Intel Core i7-4870HQ, and the memory is 16G. In the comparative experiment, two map environments were constructed, and the size, shape, and number of obstacles of different environments were different. By setting up different environments, the performance of the proposed improved planning algorithm is analyzed and verified in terms of path length, path smoothness, running time, and obstacle avoidance ability.

A. Improved A* Algorithm Verification

To verify the advantages of the improved A* algorithm in path smoothness, pathfinding efficiency, and security, this paper simulates different environments on Matlab, namely a simple environment (10*10 grid map) and a complex environment (20*20 grid map), where " △ " identifies the beginning and "○" identifies the finding. Compared with the simple environment, the number of map environments and obstacles has doubled in complex environments, and the number of searchable sub-nodes has expanded by 4 times. It can effectively evaluate the performance of the upgraded A* technique. A* algorithm shows in Fig. 7.

Fig. 7 and Fig. 8 shows that the enhanced A* algorithm can prevent too many vertices in the path planning process, and then plan a route with a shorter path length and a higher path smoothness. Fig. 9 and Fig. 10, we can see that the improved algorithm called A* will identify a shorter route in fewer seconds in a complex environment, which is obviously better than the traditional A* algorithm. The direction energy parameters of path planning for different algorithms in two different environments are shown in Table II, and the data are recorded with two decimal places:

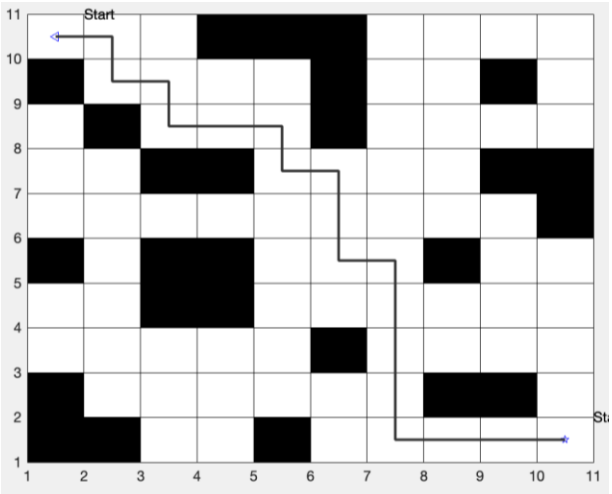

Fig. 7. Path planning of the traditional A* algorithm in a simple environment.

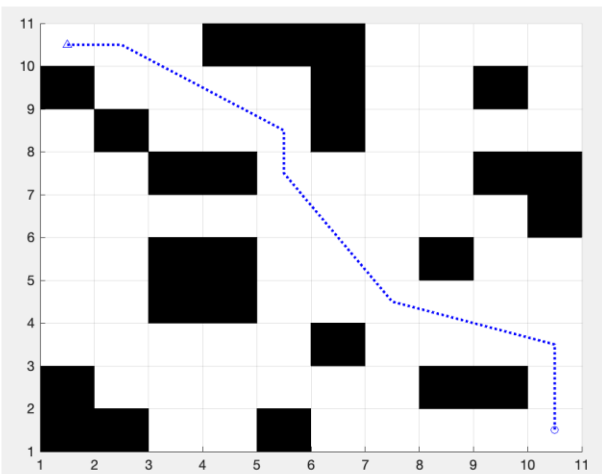

Fig. 8. Path planning of the improved A* algorithm in a simple environment.

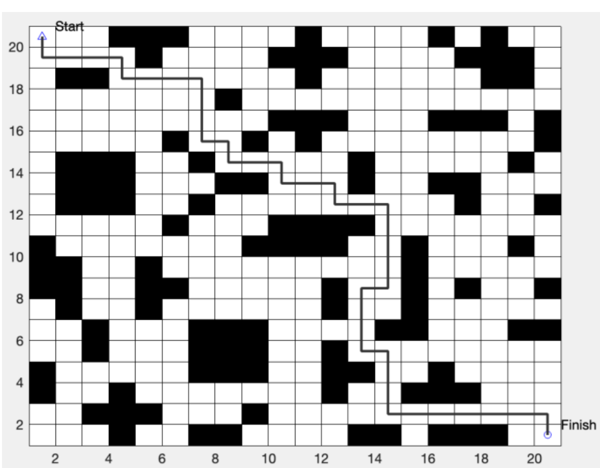

Fig. 9. Path planning of traditional A* algorithm in a complex environment.

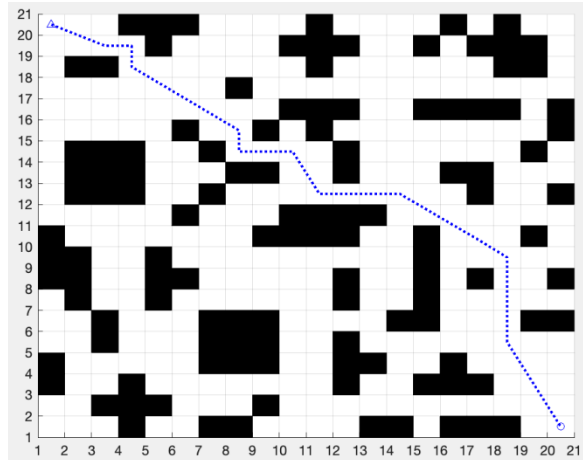

Fig. 10. Path planning of the improved A* algorithm in complex environments.

TABLE II. PARAMETERS OF PATH PLANNING IN DIFFERENT ENVIRONMENTS

		Path length	***Vertices***	***Planning time***
Simple environment	Traditional A* algorithm	18.00	10	7.69
	Improved A* algorithm	14.83	5	5.48
Complex environment	Traditional A* algorithm	38.00	16	22.89
	Improved A* algorithm	29.85	10	11.03

From this, we can see that in comparison to the typical A* algorithm, the enhanced A* method has a comprehensive reduction in the number of breakpoints by 44%, a comprehensive reduction in path length by 17.18%, and a comprehensive reduction in planning time by 14.68%. The enhanced A* method performs quite well regarding route length and roughness.

B. Fusion Obstacle Avoidance Function Verification

The A* technique's minimization evaluation has to be verified for different obstacles after integrating the obstacle avoidance ability. Construct a simulation environment 3, which is a 20*20 grid map containing movable obstacles, represented by yellow squares, whose moving coordinates are ([14, 7], [14, 11]), and contains several Unknown static obstacles represented by gray squares. Numerous factors have an impact on the efficacy as well as the efficiency of the enhanced routing technique. The experimental environment parameter settings are shown in Table III:

TABLE III. EXPERIMENTAL ENVIRONMENT PARAMETER SETTINGS

Parameter type value	
Direction angle α	0.05
Static distance β	0.2
Dynamic Distance γ	0.3
Smoothing factor σ	1
Offset term η	0.1

The following figure displays the experiment's findings:

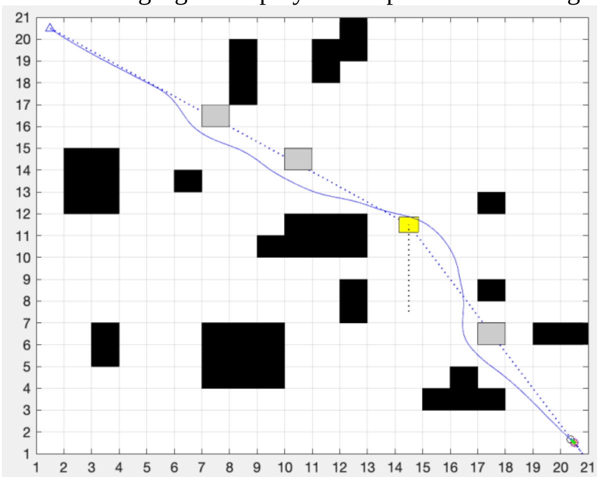

Fig. 11. Path planning of the fusion obstacle avoidance ability A* algorithm.

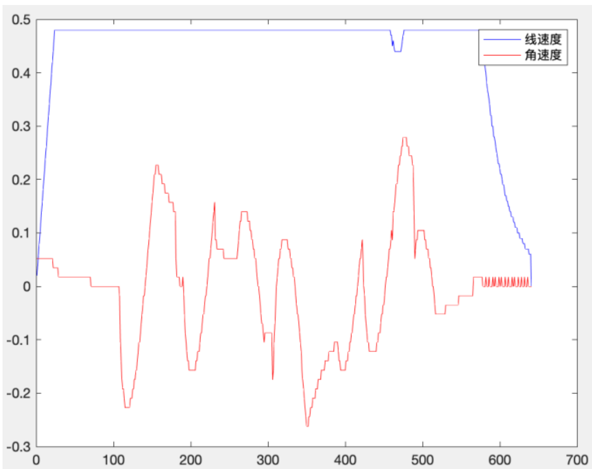

Fig. 12. The speed change diagram of the smart car.

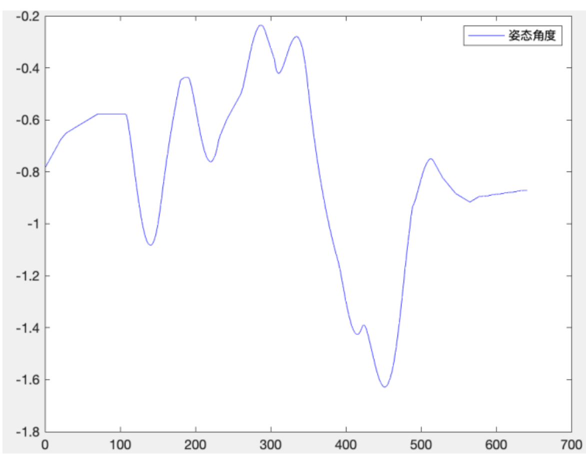

Fig. 13. The attitude angle change diagram of the smart car.

Through the analysis of Fig. 11, 12, and 13, we can conclude that for static obstacles that appear on the global planning path, when the distance between the smart car and the obstacle exceeds the calibration value, the car can change the angular velocity, thereby affecting the attitude angle. Realize automatic obstacle avoidance. When facing a dynamic obstacle, the car slows down, changes the angular velocity value, and greatly changes the attitude angle, so as not to more violent collision with the unstable obstruction. The route routing efficiency metrics for the A* approach when combined with the DWA technique is shown in Table IV.

TABLE IV. PATH PLANNING COMPARISON TABLE

	Number of inflection points	***Path length***	***Planning time***
Global planning path	1	29.85	11.28
Final path	0	38.60	60.50

From this, it is evident that, compared with the global route diagram drawn via the improved A* algorithm, the path integrated with the DWA algorithm has a higher smoothness and is more suitable for the actual production and living environment when the path length is not much different. The path realizes the optimal global path and excellent obstacle avoidance ability of the intelligent logistics vehicle.

C. Improved Algorithm Compared with Other Algorithms

The ancient route technique known as the "ant colonies program" was created by summarizing how ants forage in the wild [21-27]. The selection basis is that the shorter route has higher pheromone concentration along the route, which guides other ants to forage along the route and forms positive feedback to obtain the shortest route to find food. The ant colony algorithm can also provide global route scheduling for the fusion algorithm. The path planning of the ant colony algorithm in a complex setting (environment 2) is shown in Fig. 14:

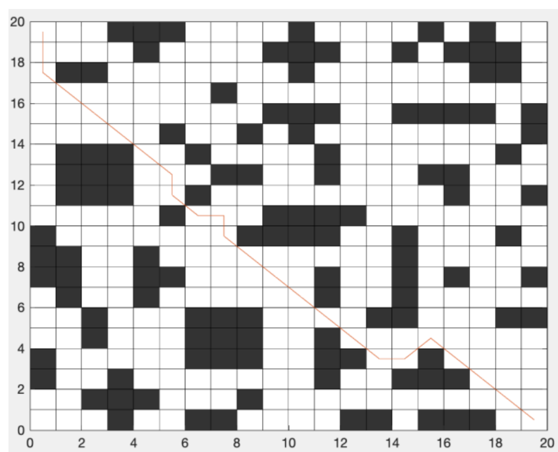

Fig. 14. Ant colony algorithm planning global path.

The path planning comparison table is shown in Table V:

TABLE V. PATH PLANNING COMPARISON TABLE OF IMPROVED A* ALGORITHM AND ANT COLONY ALGORITHM

	Path length	***Vertices***	***Planning time***
Improved A* algorithm	29.85	10	11.03
Ant Colony Algorithm	32.04	9	12.08

From the above Table V, we can analyze that throughout the procedure of developing a global route, the ant colony scheme has the disadvantages of obliquely crossing the obstacle vertices, insufficient smoothness of the intended route, and long path length. By comparison, the high smoothness and low route size of the enhanced A* algorithm in global scheduling is proved again.

VI. Conclusion and Future Work

Focusing on dynamic obstacle-avoiding route scheduling problems for storing robotics, the study put forward an A* route scheduling algorithm integrated with the DWA obstacle avoidance algorithm. Fitting greatly optimizes the path smoothness and path length of the global path planning, and realizes planning a better route in a shorter time. By integrating the DWA algorithm, it realizes automatic avoidance of static and dynamic obstacles on the planned path, improves the flexibility and the method's effectiveness at finding results in complex environments, and makes up for the shortcomings of the A* algorithm that cannot achieve dynamic obstacle avoidance.

Through MATLAB simulation experimental results, the efficacy of the enhanced method suggested in this investigation is verified in terms of route smoothness and the capacity to prevent obstacles. However, the more advanced method still has the problems of long planning time and poor adaptive proficiency of the integration technique. These are the routes for further research on the path planning of smart cars.

VII. Future Work Directions

In our future research work, we elaborate on an efficient global planning algorithm that can be designed and can be real-time used in different real-world environments and achieve the good goal of reducing the planning time by *80%* based on the current method.

VIII. Funding Statement

This research is supported by the Shandong Provincial Undergraduate Teaching Reform Project (Grant Number: Z2021450) & the Shandong Provincial Natural Science Foundation of P.R China (Grant Number: ZR2020QF069).

Acknowledgment

The authors are grateful to the anonymous referees for their insightful comments and suggestions and Chinese authors are also very thankful to Mohammad Ali Jinnah University (MAJU), Karachi, Sindh, Pakistan who supported us in this research work.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present research paper.

References

[1] He Liming. Solidly promoting the construction of a Chinese-style modern logistics system - Review and Prospect of my country's modern logistics development in 2022 [J]. Logistics Technology and Application, 2023,28(02):42-45.

[2] Zhang Chenbeixi, Huang Zhiqiu. A review of the development of automatic guided vehicles (AGV) [J]. China Manufacturing Informationization, 2010,39(01):53-59.

[3] Wu Qiping, Jin Yaping, Ren Pingping. The key technology status and development trend of automatic guided vehicle (AGV) [J]. Manufacturing Automation, 2013,35(10):106-109+121.

[4] Borenstein J, Koren Y . Real-time obstacle avoidance for fast mobile robots[J]. IEEE Transactions on Systems Man & Cybernetics, 2002, 19(5):1179-1187.

[5] Hao W , Jie D , Wang M , et al. Research on Robot Path Planning Based on Fuzzy Neural Network Algorithm[C]// IEEE Advanced Information Technology, Electronic and Automation Control Conference.

[6] Common, Ren Yan. Robot Path Planning Based on Improved Genetic Algorithm [J]. Combined Machine Tool and Automatic Processing Technology, 2023, (02): 23-27.

[7] Ogren P , Leonard N E .A convergent dynamic window approach to obstacle avoidance[J].IEEE Trans. on Robotics, 2005, 21(2):188-195.DOI:10.1109/TRO.2004.838008.

[8] Xiong Junli, Huang Huayi. Path Planning of Mobile Robots Combined with Improved Ant Colony and DWA [J]. Mechanical Design and Manufacturing: 1-8.

[9] Dan K, Manning C D. A parsing: fast exact Viterbi parse selection[J]. Proc. HLT-NAACL, 2003, 2003.

[10] Yang Guihua, Wei Jiale. Path Planning for Logistics Robots Based on Improved A~* and DWA Algorithms [J]. Science Technology and Engineering, 2022, 22(34): 15213-15220.

[11] Doopalam T, Byambaa D, Jin L D . Hybrid Motion Planning Method for Autonomous Robots Using Kinect Based Sensor Fusion and Virtual Plane Approach in Dynamic Environments[J]. Journal of Sensors, 2015, 2015:1-13.

[12] Yang Mingliang, Li Ning. Path Planning of Mobile Robot Based on Improved A~* Algorithm [J]. Mechanical Science and Technology, 2022, 41(05): 795-800.

[13] Kala R, Shukla A, Tiwari R. Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning[J]. Artificial Intelligence Review, 2010, 33(4):307-327.DOI:10.1007/s10462-010-9157-y.

[14] Hao Xiangrong. Application and Research of A* Algorithm in Intelligent Search [D]. Xi'an University of Architecture and Technology, 2007.

[15] Li X , Liu F , Liu J ,et al.Obstacle avoidance for mobile robot based on improved dynamic window approach[J].Turkish Journal of Electrical Engineering and Computer Sciences, 2017, 25:666-676.DOI:10.3906/elk-1504-194.

[16] Fox D, Burgard W, Thrun S. The Dynamic Window Approach to Collision Avoidance[J]. IEEE Robotics & Automation Magazine, 2002, 4(1):23-33.DOI:10.1109/100.580977.

[17] Seder,Marija; Petrović,Ivan.Dynamic window-based approach to mobile robot motion control in the presence of moving obstacles[C]//IEEE International Conference on Robotics & Automation.0[2023-06-19].DOI:10.1109/ROBOT.2007.363613.

[18] Demeester E , Nuttin M , Vanhooydonck D , et al. Global dynamic window approach for holonomic and non-holonomic mobile robots with arbitrary cross-section[C]// IEEE/RSJ International Conference on Intelligent Robots & Systems. IEEE, 2005.

[19] Chris Clark, “Autonomous Robot Navigation”,2011. https://www.cs.princeton.edu/courses/archive/fall11/cos495/COS495-Lecture5-Odometry.pdf.

[20] Li X, Liu F, Liu J, et al. Obstacle avoidance for mobile robots based on improved dynamic window approach[J]. Turkish Journal of Electrical Engineering and Computer Sciences, 2017, 25:666-676.

[21] Duan Haibin, Wang Daobo, Zhu Jiaqiang, etc. Progress in Ant Colony Algorithm Theory and Application Research [J]. Control and Decision Making, 2004(12): 1321-1326+1340. DOI: 10.13195/j.cd.2004.12. 1. duanhb.001.

[22] Tahir, Muhammad, et al. "The Novelty of A-Web based Adaptive Data-Driven Networks (DDN) Management & Cooperative Communities on the Internet Technology." Int. J. Adv. Comput. Sci. Appl 8 (2017): 16-24.

[23] Zheng, Xiao, et al. "Computational Analysis based on Advanced Correlation Automatic Detection Technology in BDD-FFS System." International Journal of Advanced Computer Science and Applications 13.8 (2022).

[24] Liu, Zhouqi, et al. "An Improved Poisson Surface Reconstruction Algorithm based on the Boundary Constraints." International Journal of Advanced Computer Science and Applications 14.1 (2023).

[25] Huang, Jin, et al. "The Effective 3D MRI Reconstruction Method Driven by the Fusion Strategy in NSST Domain." International Journal of Advanced Computer Science and Applications 14.4 (2023).

[26] Ahmad, Zeeshan, Muhammad Tahir, and Iftikhar Ali. "Analysis of beamforming algorithms for antijams." 2013 XVIIIth International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED). IEEE, 2013.

[27] Ahmad, Zeeshan, and Muhammad Tahir. "Named data networking (NDN) new approach to future Internet architecture design: A survey." Int. J. Informat. Commun. Technol. 2.3 (2013): 155-164.