applied sciences

MDPI

Article

Based on the Integration of the Improved A* Algorithm with the Dynamic Window Approach for Multi-Robot Path Planning

Yong Han, Changyong Li * and Zhaohui An

School of Mechanical Engineering, Xinjiang University, Urumqi 830039, China; 107552204183@stu.xju.edu.cn (Y.H.); 107552204145@stu.xju.edu.cn (Z.A.)
* Correspondence: lcy@xju.edu.cn

Abstract: With the escalating demand for automation in chemical laboratories, multi-robot systems are assuming an increasingly prominent role in chemical laboratories, particularly in the task of transporting reagents and experimental materials. In this paper, we propose a multi-robot path planning approach based on the combination of the A* algorithm and the dynamic window algorithm (DWA) for optimizing the efficiency of reagent transportation in chemical laboratories. In environments like chemical laboratories, dynamic obstacles (such as people and equipment) and transportation tasks that demand precise control render traditional path planning algorithms challenging. To address these issues, in this paper, we incorporate the cost information from the current point to the goal point into the evaluation function of the traditional A* algorithm to enhance the search efficiency and add the safety distance to extract the critical points of the paths, which are utilized as the temporary goal points of the DWA algorithm. In the DWA algorithm, a stop-and-wait mechanism and a replanning strategy are added, and a direction factor is included in the evaluation function to guarantee that the robots can adjust their paths promptly in the presence of dynamic obstacles or interference from other robots to evade potential conflicts or traps, thereby reaching the goal point smoothly. Additionally, regarding the multi-robot path conflict problem, this paper adopts a dynamic prioritization method, which dynamically adjusts the motion priority among robots in accordance with real-time environmental changes, reducing the occurrence of path conflicts. The experimental results highlight that this approach effectively tackles the path planning challenge in multi-robot collaborative transportation tasks within chemical laboratories, significantly enhancing transportation efficiency and ensuring the safe operation of the robots.

Keywords: A* algorithm; DWA algorithm; multi-robot path planning

Academic Editor: Pedro Couto

Received: 14 November 2024
Revised: 18 December 2024
Accepted: 3 January 2025
Published: 4 January 2025

Citation: Han, Y.; Li, C.; An, Z. Based on the Integration of the Improved A* Algorithm with the Dynamic Window Approach for Multi-Robot Path Planning. *Appl. Sci.* **2025**, *15*, 406. https://doi.org/10.3390/app15010406

1. Introduction

With the swift advancement of robotics, multi-robot systems (MRS) are increasingly being employed in complex and dynamic settings [1]. In environments such as chemical laboratories, the conventional task of transporting reagents is typically carried out manually, which is not only labor-intensive but also poses safety risks, especially when dealing with hazardous chemicals. With the proliferation of automation technology, the utilization of robots to undertake the transportation of reagents and materials instead of manual labor not only enhances efficiency but also effectively mitigates the occurrence of human operational errors and safety incidents. Chemical laboratories frequently encounter complex environmental circumstances of confined spaces, dynamic obstacles, and fixed obstacles, which pose challenges for robot path planning. To ensure that the robots can

complete the transportation tasks safely and promptly, how to strike an effective balance between global path planning and local obstacle avoidance becomes a key issue in multi-robot path planning. Common robot path planning algorithms encompass global path planning [2] and local path planning [3]. Global path planning algorithms include the Dijkstra algorithm [4], the A* algorithm [5,6], and the ant colony algorithm [7,8], et cetera, and local paths incorporate the dynamic window method [9,10], the artificial potential field approach [11,12], and the time elastic band (TEB) method [13,14].

In the domain of path planning, numerous scholars have carried out profound and extensive research. The A* algorithm is the prevalent global path planning algorithm; however, the paths it generates exhibit poor smoothness and might engender a considerable number of redundant computations. Juntao Yan [15] et al. guaranteed the feasibility of the path by incorporating a turning factor into the heuristic function and optimizing the search neighborhood. Yawen Dong [16] broadened the search scope and established a dual-layer position-guided information evaluation function, thereby achieving a smoother path. Through integrating the obstacle rate coefficient into the evaluation function of the A* algorithm, an adaptive cost function is constructed, and by enhancing the search approach, the search efficiency and path safety are enhanced [17]. When a robot applies the dynamic window approach for path planning in navigation tasks, it may become stuck in a local optimum solution and be incapable of finding the optimal path. Xu Chi [17] integrated A* with DWA, leveraging the global path planning capabilities of the A* algorithm and the local obstacle avoidance capacity of DWA, thereby overcoming the limitations of a single approach and improving the robot's ability to navigate through complex environments. A flexible dynamic window technique employing a fuzzy controller for optimizing the dynamic obstacle avoidance of mobile robots was put forward, which allows the robot to move towards the target as rapidly as possible [18]. Through the introduction of a designed risk function into the evaluation function of the standard DWA (dynamic window approach), the probability of a collision between dynamic obstacles and the robot can be assessed, allowing the robot to effectively evade faster obstacles [19]. Regarding multi-robot path planning. Zhang C [20] initially employed the RRT* algorithm to acquire the shortest path and subsequently utilized an advanced particle swarm optimization algorithm to obtain the time schedules for multiple robots, effectively resolving conflicts. Daojin Yao [21] proposed a hierarchical distributed multi-AGV path planning algorithm, improving the combination of the A* and DWA algorithms and introducing a cooperative planning strategy to reduce the probability of conflicts between mobile robots. A multi-robot dual-layer planning algorithm featuring an improved A* algorithm and a conflict resolution strategy was put forward. In the first layer, the improved A* algorithm was used to obtain the initial path for a single robot. In the second layer, a time dimension was introduced to anticipate robot conflicts [22]. A safe A* algorithm was developed with a technique for identifying key path points to guarantee smooth trajectory following, and real-time obstacle avoidance was achieved through adaptive window-based motion planning [23]. The paper [24] presents a new path planning strategy combining an optimized A* algorithm for global path planning, adaptive navigation and obstacle avoidance for local planning, and a collision recognition and task prioritization strategy for coordinating multiple robots.

Although the existing literature has extensively explored robot path planning, particularly focusing on obstacle avoidance strategies, our research emphasizes the issue of the "path deadlock" in dynamic environments. Specifically, it addresses how to re-plan the path in real time when unknown obstacles block the original route. While most studies focus on obstacle avoidance, research on path re-planning remains relatively limited. To tackle this challenge, we propose an innovative path re-planning method tailored to the specific application scenario of chemical laboratories. This method incorporates a

parking–waiting strategy and a multi-robot priority scheduling mechanism, effectively resolving the issues of path blockage and motion conflicts within multi-robot systems while successfully handling path obstructions in dynamic environments. Regarding the improvement and analysis of the aforementioned algorithms, a novel multi-robot path planning method is proposed, combining the A* algorithm and the dynamic window approach (DWA). Within the A* algorithm, we adopted a dynamic evaluation function that adapts based on distance cost weighting. This modification enhanced the algorithm's efficiency in complex environments and enabled the dynamic adjustment of the weights in response to diverse environmental variations, thereby augmenting the flexibility and robustness of the path planning. Furthermore, a redundant node deletion strategy was introduced to further optimize the efficiency of path calculation, avoiding unnecessary computations and path redundancies and enhancing the overall planning performance. In the section of the dynamic window method, we expanded the evaluation function to incorporate an adaptive adjustment mechanism for the heading angle. This tactic dynamically modifies the robot's heading angle, enabling the path planning to adapt more precisely to various driving environments and minimizing the trajectory deviation caused by unreasonable heading, thereby enhancing the precision and motion efficiency of navigation. To further enhance the adaptability of the algorithm, this study also put forward a parking-and-waiting strategy, which can promptly halt and wait when encountering unforeseen obstacles, thereby evading unnecessary collision risks; meanwhile, the path replanning strategy guarantees that if an infeasible path emerges during the robot's execution, a new feasible path can be rapidly identified, preventing prolonged system stagnation. In the context of multi-robot path planning, we introduced a dynamic priority strategy for addressing the obstacle avoidance issue among robots. This strategy effectively precludes conflicts and deadlocks among robots and guarantees that multiple robots can collaborate in shared spaces without colliding or delaying tasks. By dynamically regulating the priorities between robots, we can ensuring that multiple robots can complete tasks efficiently and stably in complex and dynamically evolving environments. In conclusion, this approach significantly enhances the safety, efficiency, and adaptability of multi-robot path planning through innovative optimizations in various modules.

2. Improvements to the A* Algorithm

2.1. Traditional A Algorithm*

The A* algorithm is a heuristic technique employed to identify the shortest path. The initial node serves as the starting point, expanding to the surrounding domain and selecting the node with the lowest cost after evaluation through a heuristic function. With this node, the expansion continues around, and subsequently, the node with the lowest cost is selected. The above steps are repeated until the selected node coincides with the goal point. The cost function employed in the A* algorithm is expressed as follows:

$$f(u) = g(u) + h(u) \tag{1}$$

Let u represent the current node. The term $f(u)$ denotes the total cost from the starting point to the goal. Additionally, $g(u)$ indicates the actual cost incurred from the starting point to the current node, while $h(u)$ estimates the cost from the current node to the goal. The efficiency and search quality of the A* algorithm largely rely on the selected heuristic function. Common heuristic functions for the distance functions used in the A* algorithm include the Manhattan, Euclidean, and Chebyshev functions, which are denoted as follows. The Euclidean distance yields the paths of the best quality. Therefore, the heuristic function formula utilized in this paper is the Euclidean distance function. In Equations (2) to (4), lc

and lg represent the x-coordinate and y-coordinate of the starting point, respectively, while tc and tg represent the x-coordinate and y-coordinate of the endpoint, respectively.

$$f(u) = |lc - tc| + |lg - tg| \tag{2}$$

$$f(u) = \sqrt{(lc - tc)^2 + (lg - tg)^2} \tag{3}$$

$$f(u) = |lc - tc| + |lg - tg| + (\sqrt{2} - 2)\min(lc - tc, lg - tg) \tag{4}$$

2.2. Improvement to the A* Algorithm

The paths of traditional A* algorithms typically contain numerous redundant nodes, leading to increased path lengths and prolonged planning times, and in complex environments, even passing through obstacles. Hence, improvements to the A* algorithm are requisite to enhance the search efficiency.

2.2.1. Improvements in the Evaluation Function

The evaluation function of the traditional A* algorithm lacks adaptability to dynamic changes or the complexity of the environment and is unable to flexibly adjust the search strategy to cope with different situations, resulting in limited effectiveness in complex environments. Therefore, the evaluation function is weighted. When the weight coefficient is large, the A* algorithm will rapidly expand towards the target area, but there is a possibility of missing the best route. In the case where the weight coefficient is not large, the method is more likely to find the best path, but its search speed will be reduced. Therefore, this paper introduces the concept of distance cost into the traditional heuristic function, enabling the heuristic function to adaptively adjust according to the current relative distance between the robot and the target. The optimized heuristic function is denoted as:

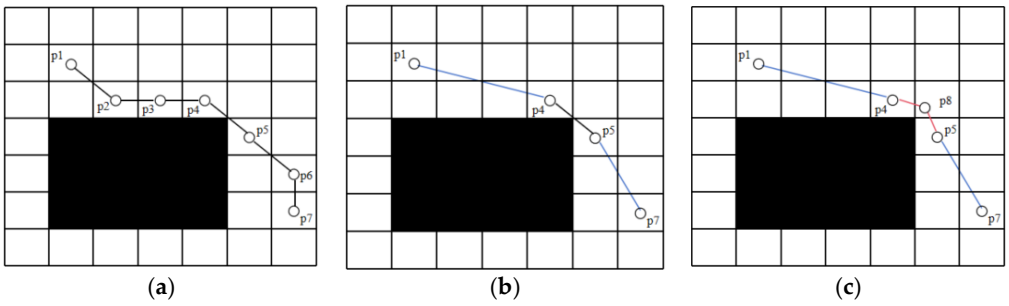
$$f(u) = g(u) + \ln(D/H + e^2) * h(u) \tag{5}$$

The value D represents the distance from the current position to the goal, while H indicates the distance from the starting point to the current position.

2.2.2. Selection of Critical Points

Traditional A* algorithms frequently generate a considerable quantity of redundant nodes and avoidable corners in path planning. This not only results in insufficiently smooth paths and an increase in the path length, but the path may even pass through obstacles, thereby influencing the motion efficiency and performance of robots in complex environments. Hence, eliminating these redundant nodes constitutes a critical step in path optimization. Through path optimization, the robot can move in a smoother and more efficient manner, thereby enhancing the overall operational performance and environmental adaptability. This optimization not only alleviates the computational load but also significantly enhances the robot's navigational accuracy. The optimized path is presented in Figure 1.

As shown in Figure 1a, the black line segments represent the robot's initial path. In Figure 1b, the blue line represents the path after the first optimization, and in Figure 1c, the red line represents the path after the second optimization.



(**a**) (**b**) (**c**)

Figure 1. Identification of key points. (**a**) The A*-derived initial path. (**b**) The initial path. (**c**) The second path.

First, the A* algorithm is employed to generate the initial paths p1, p2...p7 and to record all the nodes passed on the path, encompassing redundant nodes and turning points. These nodes represent the planned trajectory of the robot from the origin point to the goal point. In the next step, each node pi along the path is traversed, and its front and rear neighboring nodes pi − 1 and pi + 1 are examined. For this purpose, the vector from *pi* to *pi* − *1* and the vector from *pi* to *pi* + *1* are computed, and the angle between them is computed by the dot product and the modulus length of these two vectors θ. Specifically, the angle between the two vectors θ can be calculated by the following formula:

$$\cos\theta = \frac{\overrightarrow{pipi-1} \cdot \overrightarrow{pipi+1}}{\left|\overrightarrow{pipi-1}\right| \cdot \left|\overrightarrow{pipi+1}\right|} \tag{6}$$

If the angle θ is equal to 0, then pi, pi − 1, and pi + 1 are collinear. In this case, node pi can be regarded as redundant and can be deleted, and its parent node is modified to connect both pi − 1 and pi + 1. If the pinch angle is not 0, the linear equations of pipi − 1 and pipi + 1 are established, respectively. The safety distance f is introduced; the value d denotes the minimum distance between the obstacle point and the straight line. If d $\leq$ f, there is an obstacle between the two points, and then the current point is set as the parent node; otherwise, if there is no obstacle between the two points, the intersection of the two straight lines is the parent node, and other redundant nodes are deleted. The above steps are repeated. As shown in Figure 1, in the first optimization, nodes p2, p3, and p6 are deleted, and p1p4 and p5p7 are connected because the line of p4p7 passes through the obstacle, so p4 is retained. The second optimization extends p1p4 and p5p7 to intersect at point p8, which is the new parent node. The final path is p1p8p7.

3. Dynamic Window Approach

The dynamic window approach (DWA) is a well-established method used for local path planning. It enables robots to avoid obstacles and move towards a goal in complex environments by calculating feasible velocities and accelerations in real time.

3.1. Robot Modeling

The dynamic window method forecasts the travel path over a specific time interval at these velocities by sampling certain combinations of the velocity values in the velocity space. The paths are evaluated collectively, and the velocity of the highest-ranked path is chosen to control the robot's motion. This paper focuses on a differential-speed mobile robot, with an analysis of its motion model provided in Equation (7).

$$\begin{cases} x_t = x_{t-1} + v_{t-1} \triangle t \cos(\theta_t) \\ y_t = y_{t-1} + \nu_{t-1} \triangle t \sin(\theta_t) \\ \theta_t = \theta_{t-1} + w_{t-1} \triangle t \end{cases} \tag{7}$$

In this equation, v_{t-1} refers to the robot's linear velocity at time $t-1$, ω_{t-1} refers to the angular velocity at time $t-1$, and x_t, y_t, and θ_t represent the robot's position at time t. $\triangle t$ represents the time difference between $t-1$and t. θ_t represents the angle rotated within time interval $\triangle t$. x_{t-1}, y_{t-1}, and θ_{t-1} represent the robot's position at time $t-1$.

3.2. Velocity Sampling

As a result of the restrictions in the mobile robot's hardware and environmental factors, there exists a range of limits in the speed sampling space of the mobile robot.

1. Linear and angular velocity constraints for robots.

$$V_s = \{(v, \omega) | v_{\min} \leq v \leq v_{\max}, \omega_{\min} \leq \omega \leq \omega_{\max}\} \tag{8}$$

where $v_{\min}$ is the minimum linear velocity, $v_{\max}$ indicates the maximum possible linear velocity, $\omega_{\min}$ represents the minimum possible angular velocity, and $\omega_{\max}$ denotes the maximum angular velocity.

2. Motor performance effects and acceleration constraints.

$$V_d = \left\{(v, \omega) \middle| v_c - \dot{v}_b \Delta t \leq v \leq v_c + \dot{v}_a \Delta t,\ \omega_c - \dot{\omega}_b \Delta t \leq \omega \leq \omega_c + \dot{\omega}_a \Delta t\right\} \tag{9}$$

Here, v_c represents the robot's linear speed, ω_c refers to its angular speed, while $\dot{v}_a$ indicates the maximum linear acceleration, and $\dot{\omega}_a$ denotes the maximum angular acceleration.

3. Impact of obstacles.

$$V_a = \left\{(v, \omega) \middle| v \leq \sqrt{2dist(v, \omega)\dot{v}_b},\ \omega \leq \sqrt{2dist(v, \omega)\dot{\omega}_b}\right\} \tag{10}$$

The maximum linear deceleration is represented by $\dot{v}_b$, $\dot{\omega}_b$ denotes the maximum angular deceleration, and $dist(v, \omega)$ indicates the distance from the robot's current velocity to the closest obstacle in its path.

By combining the three types of velocity constraints mentioned above, the final velocity sampling space of the robot exists at the convergence of the three velocity spaces.

$$V_r = V_s \cap V_d \cap V_a \tag{11}$$

3.3. Evaluation Function

After sampling the velocity space, a kinematic model based on the robot predicts multiple possible motion trajectories. Subsequently, these trajectories are evaluated, and the best motion trajectory is selected, and the robot moves forward according to the velocity of the best trajectory. In order to enable the robot to reach the goal point successfully, an adaptive heading factor c is introduced, enabling the robot to be directed more precisely towards the target while minimizing the risk of collisions. The evaluation function associated with the DWA algorithm is as follows:

$$G(v, w) = \sigma(a * c * heading(v, w) + \beta * dist(v, w) + \gamma * velocity(v, w)) \tag{12}$$

where $heading(v, w)$ represents the function that evaluates the azimuth, $dist(v, w)$ signifies the distance to the nearest obstacle on the map at the endpoint of the predicted path, and $velocity(v, w)$ stands for the robot's current linear velocity. The terms α, β, γ, and σ are the

weights for each corresponding factor. As in Equation (13), λ is the adjustment coefficient, θ_{cur} denotes the robot's current heading angle, and θ_{des} is the target orientation angle. y_{goalt} is the vertical coordinate of the robot's temporary target point in the next time frame, x_{goalt} is the horizontal position of the robot's provisional target at the next moment, and v_x and v_y represent the robot's sideways displacement.

$$c = 1 + \lambda \cdot |\theta_{\text{des}} - \theta_{cur}| \tag{13}$$

$$\theta_{cur} = \arctan(v_y, v_x) \tag{14}$$

$$\theta_{des} = \arctan(y_{goalt} - y_t, x_{goalt} - x_t) \tag{15}$$

3.4. Improvement of the DWA Algorithm

When the robot and the dynamic obstacle are in close proximity to each other or their trajectories intersect at the same point at a certain moment, the robot may opt to take a longer detour in order to avoid the dynamic obstacle, resulting in a deviation from the original heading. This behavior not only lowers the robot's path planning accuracy but also may have an impact on its overall operational efficiency. Therefore, in order to effectively address this issue, an improved dynamic window method is proposed, which introduces the concept of a "stop-and-wait state". In this state, the robot halts its advancement and gives priority to the passage of moving obstacles, thereby avoiding unnecessary yawing caused by sharp turns or detours.

The improved algorithm constrains the range detection of the robot by setting distance and angle thresholds to ensure that the robot can respond in a timely manner when the obstacle is close and there is a risk of collision. Specifically, the distance threshold d = 2 and the angle threshold θ = 45° are set as judgment conditions, as shown in Figure 2. When a moving obstacle enters this detection range, the robot automatically enters the stop-and-wait state, halts its advancement and waits for the obstacle to pass, thereby reducing the risk of collision.

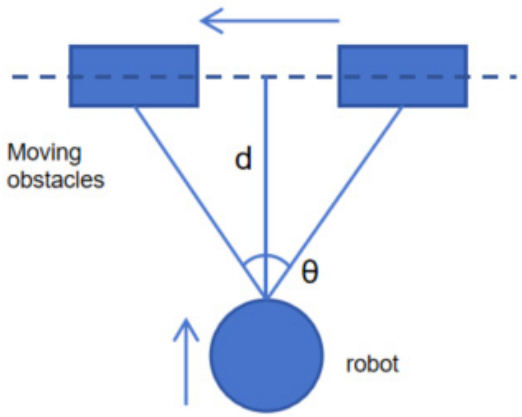
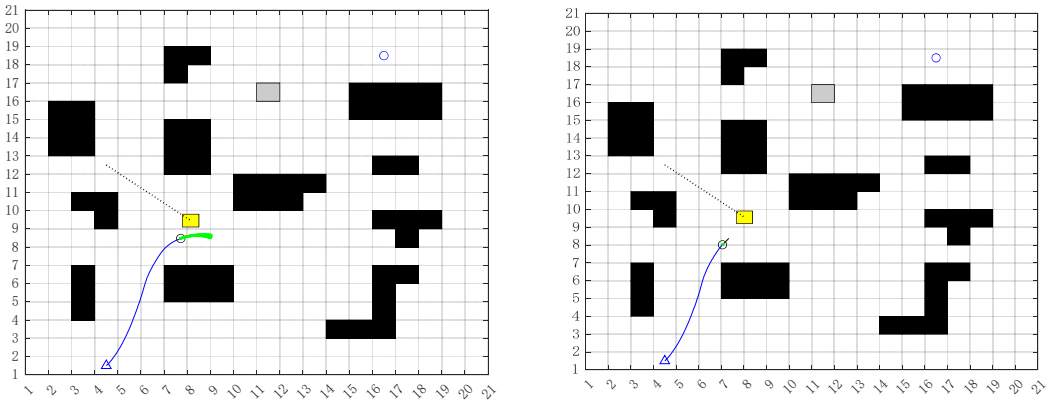


Figure 2. Parking-and-waiting strategy.

As shown in Figures 3a,b and 4a,b, the black squares represent known static obstacles, the gray squares represent unknown static obstacles, the yellow squares represent dynamic obstacles, the blue triangle represents the robot's starting point, and the blue circle represents the robot's endpoint. As shown in Figure 3a, when the robot and the dynamic obstacle coincide with the path point at a certain moment, the traditional dynamic window method will deviate from the original heading and go further and further in order to bypass the dynamic obstacle. However, in the enhanced dynamic window approach shown in

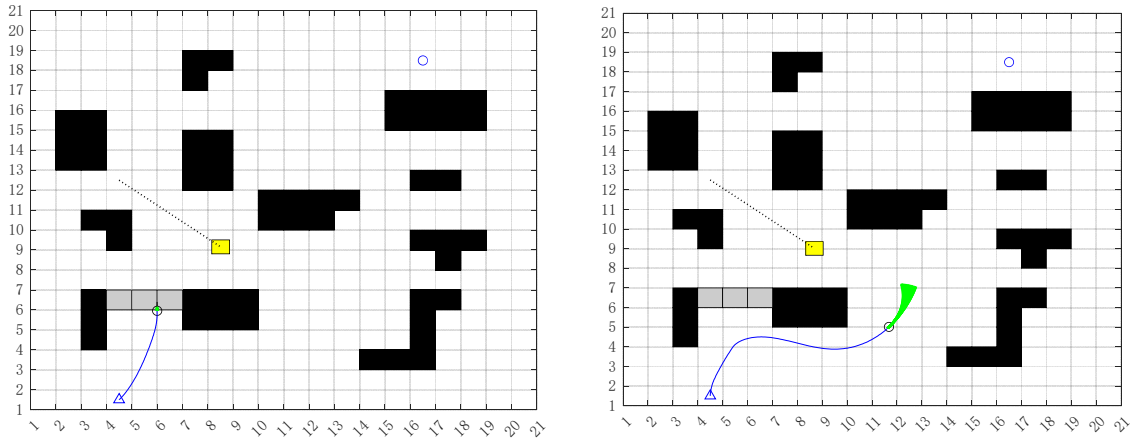
Figure 3b, the robot will halt and wait until the moving obstacle passes, and then the robot initiates movement.



(**a**) Conventional DWA algorithm (**b**) Improved DWA algorithm

Figure 3. Algorithm comparison.

Due to the limitations of the traditional dynamic window approach (DWA) in handling complex environments, especially when encountering U-shaped obstacles, the robot often falls into a local optimal solution, as shown in Figure 4a. For this purpose, this study designed an improved algorithm that utilizes the current position information of the robot along with the direction angle and simultaneously combined it with ultrasonic sensors to identify possible trap regions by predicting the grid nodes in the forward direction of the robot and detecting whether these nodes are occupied by obstacles.



(**a**) Conventional DWA algorithm (**b**) Improved DWA algorithm

Figure 4. U-shaped environment.

By identifying and circumventing these local optimal solutions, the robot is capable of replanning its path, as shown in Figure 4b, thereby enhancing the global optimality and operational efficiency of the overall path planning. The application of this method in complex dynamic environments demonstrates stronger adaptability, especially in the face of dense or irregular obstacle environments, which helps to greatly increase the stability and versatility of the robot's navigation system.

4. Multi-Robot Collaborative Planning

The primary challenge in multi-robot cooperative planning is to efficiently resolve path conflicts and ensure effective obstacle avoidance among the robots. Within challenging

and dynamic environments, robots not only have to accomplish their respective tasks but also have to adjust their paths in real time to avoid conflicts with other robots or obstacles.

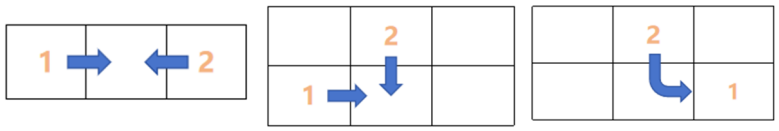
4.1. Types of Conflict

In multi-robot path planning, the principal types of conflicts that occur include node conflicts, in-phase conflicts, and occupancy conflicts.

4.2. Priority Obstacle Avoidance

In multi-robot path planning, the prioritized obstacle avoidance strategy is a core approach intended for resolving potential path conflicts, task coordination, and obstacle avoidance issues in multi-robot systems. The strategy guarantees the efficiency and safety of the system by assigning distinct priorities to each robot and forming an execution sequence based on task priorities. During actual execution, the robots determine when to execute tasks, how to avoid other robots, and how to coordinate their movements efficiently based on their respective priorities, thereby avoiding or resolving conflicts and ensuring the smooth operation of the system.

1. Collision course. As shown in Figure 5a. When two robots are traveling in opposite directions on a path and there is a possibility of collision at a certain location, the robot with higher priority will be granted precedence in passing through, and the robot with lower priority will need to temporarily stop and wait. This typically occurs in high-conflict areas such as narrow passages or intersections. To prevent a collision, the lower-priority robot may need to opt for temporarily stopping, backing up, or taking an alternate path.



(**a**) Collision course (**b**) Clash of nodes (**c**) Path deadlock

Figure 5. Types of conflict.

2. Clash of nodes. As shown in Figure 5b. When multiple robots plan to travel to the same or adjacent nodes, a competitive state might arise, especially when these robots plan to reach the same goal location simultaneously in a limited space. To address this conflict, robots with higher priority will be granted priority to pass through, while robots with lower priority need to remain in place and wait until the higher-priority robot completes its task or passes through the node.
3. Path deadlock. As shown in Figure 5c. Deadlocks are a common issue in multi-robot systems, especially during path planning and execution among robots, which occur when the robots mutually block each other, making it impossible to proceed. This typically occurs in path planning that lacks sufficient flexibility. To avoid deadlocks, the prioritized obstacle avoidance strategy takes the possibility of deadlocks into account in the planning stage and reduces the mutual interference among robots by assigning reasonable paths to robots with different priorities.
4. Reprogramming mechanisms. The replanning process is initiated when the system detects that a low-priority robot is trapped in a prolonged waiting or deadlock state. At this point, the low-priority robots recalculate their paths in accordance with the system's dynamic environment and identify new feasible paths to bypass the conflict region. The high-priority robots continue to follow the original path to carry out their tasks without being influenced. This mechanism can effectively prevent a long-term

stagnation state of the system and ensure that the tasks of the high-priority robots can be accomplished successfully.

5. Fusion Algorithm

By incorporating the advanced A* algorithm into the enhanced DWA framework, the robot can maintain the optimization of the global path in dynamic and complex environments and simultaneously demonstrate the ability to react to changes in the environment instantaneously. The global path generated by the A* algorithm is divided into segments, each representing a local goal point through a series of key waypoints, and the processing of each local goal point is taken over by the DWA algorithm, enabling the robot to avoid obstacles by predicting and dynamically adjusting its speed and direction to guarantee that the obstacle avoidance process does not deviate from the global path. In multi-robot cooperative planning, each robot can also be dynamically assigned a priority, and when the path is in conflict, the priority mechanism determines which robot goes first, ensuring the stability and efficiency of the complete system through real-time scheduling and replanning of the path. The detailed implementation of the fusion algorithm is shown in Figure 6.

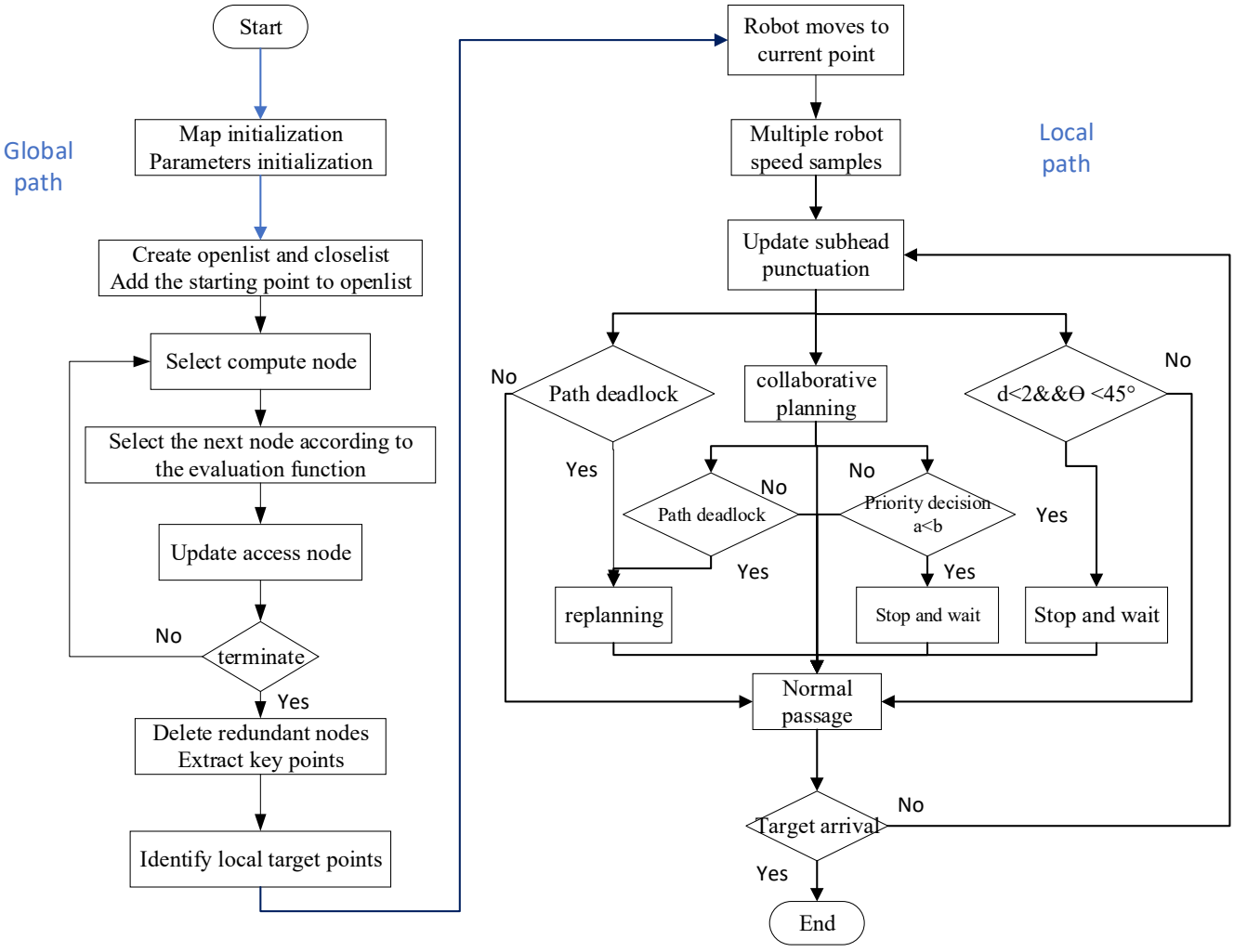


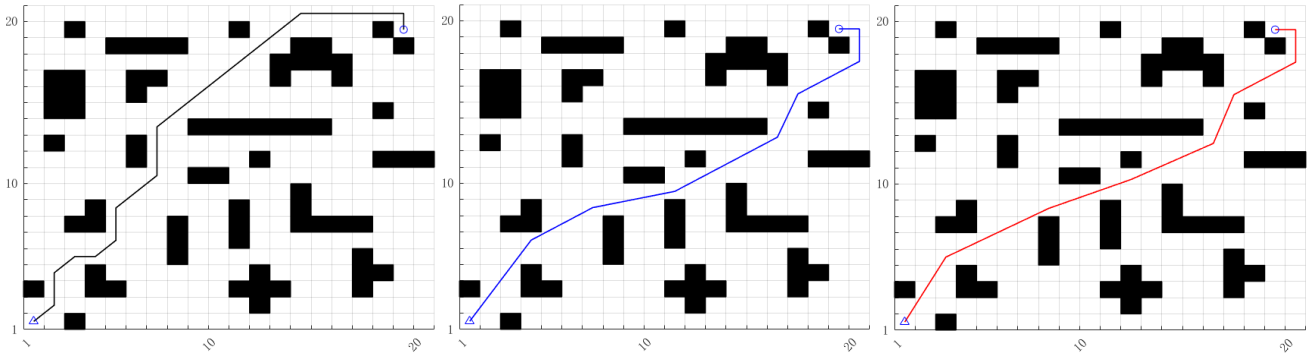
Figure 6. Flowchart of the fusion algorithm.

6. Simulative Experiments and Analysis

6.1. Improved A Algorithm Simulation Experiment*

To evaluate the performance and viability of the modified A* algorithm proposed in this paper, simulation experiments were executed within the MATLAB 2021b environment.

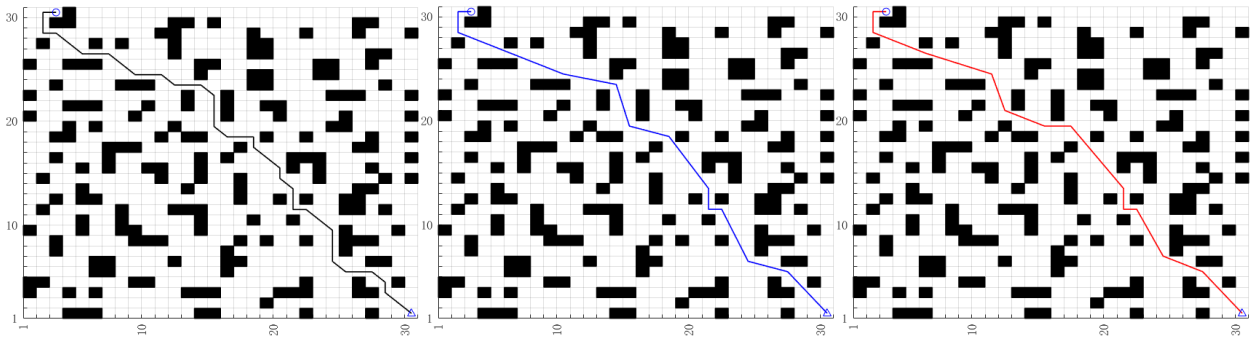
The experimental scenarios selected were 20 × 20, 30 × 30, and 50 × 50 grid maps, and the effectiveness of the enhanced A* algorithm presented in this paper was compared to that of the classic A* algorithm and algorithms from the literature [25]. In the grid maps, each grid has a size of 1 m, where black represents obstacles and white indicates traversable areas. The triangle symbolizes the robot's initial position, and the circle represents the goal point. These map scenarios of diverse scales can effectively assess the path planning capabilities of each algorithm in different complex environments. The simulation results are presented in Figure 7.



(**a**) (**b**) (**c**)

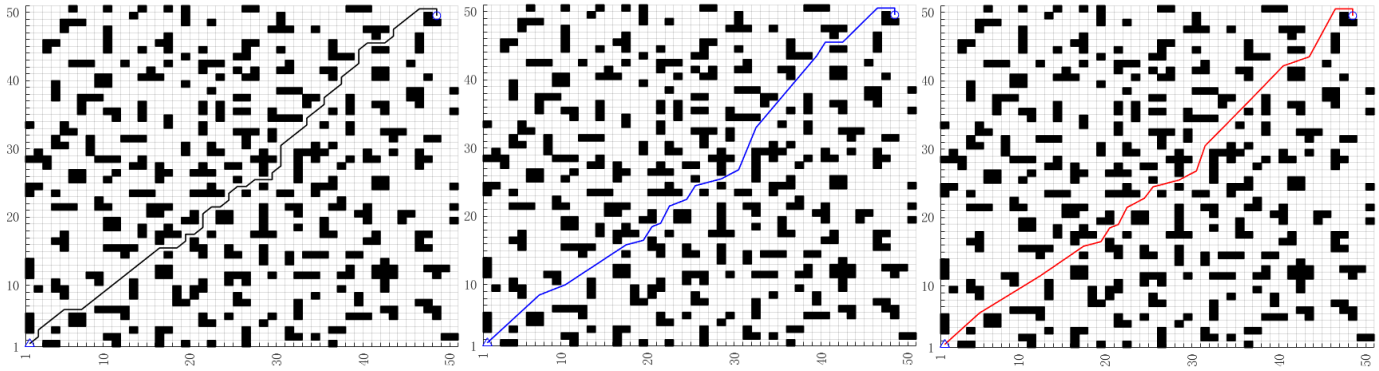
Figure 7. Comparison of simulation results for 20 × 20 raster maps. (**a**) Traditional A* algorithm. (**b**) Algorithm from [25]. (**c**) Algorithm in this paper.

By examining Figures 7a–c, 8a–c and 9a–c and Table 1, it becomes evident that the traditional A* algorithm had more turning and redundant points compared to the improved A* algorithm, which generally had fewer points. In this paper, the search efficiency was enhanced by dynamically adjusting the weight of the heuristic function in the standard A* algorithm and by carefully selecting key points. This resulted in the ability to find shorter paths more effectively and complete the search in less time, thus significantly improving the overall performance. These improvements were demonstrated in maps of various sizes. The improved A* algorithm presented in this paper achieved a reduction in path length of 6.63% and 0.63%, respectively, relative to the conventional A* algorithm and the algorithm discussed in the literature [25], respectively, and it reduced the search time by an average of 56% and 23.8% compared with the traditional A* algorithm and the algorithm from the literature [25], respectively. Meanwhile, with the increasing complexity and size of the maps, the corresponding gap between the path lengths and the search times became increasingly larger.



(**a**) (**b**) (**c**)

Figure 8. Comparison of simulation results for 30 × 30 raster maps. (**a**) Traditional A* algorithm. (**b**) Algorithm from [25]. (**c**) Algorithm in this paper.



(**a**) (**b**) (**c**)

Figure 9. Comparison of simulation results for 50 × 50 raster maps. (**a**) Traditional A* algorithm. (**b**) Algorithm from [25]. (**c**) Algorithm in this paper.

Table 1. Simulation results of three map sizes with different algorithms.

Map Size	Algorithm	Path Length (m)	Time (s)
20 × 20	Traditional A*	30.97	0.52
		29.03	0.32
		29.02	0.27
30 × 30	Algorithm from [25]	48.63	0.99
		45.73	0.58
		45.34	0.41
50 × 50	Algorithm in this paper	77.08	1.71
		72.62	0.86
		72.05	0.63

6.2. Fusion DWA Algorithm Simulation Experiment

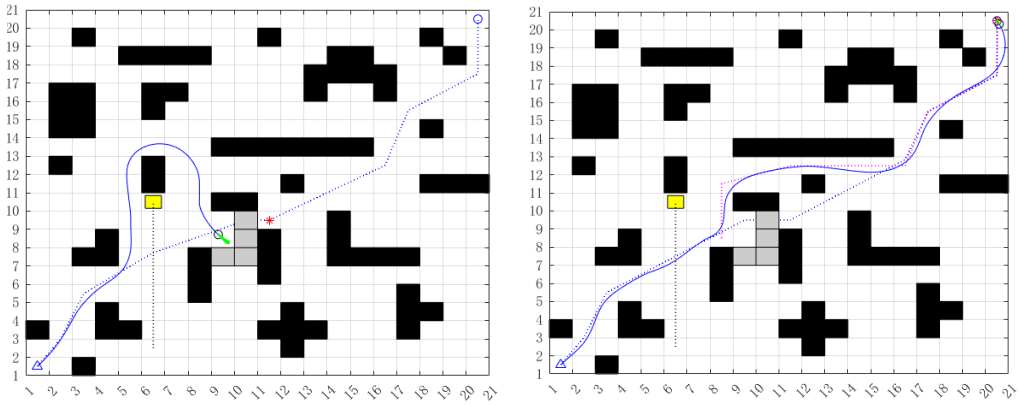
In order to assess the performance of the algorithm introduced in this study, experiments were carried out using a 20 m × 20 m grid map. In the experiment, four unidentified static obstacles were placed on the map, and dynamic obstacles were incorporated to simulate a complex dynamic environment for further examination of the algorithm's obstacle avoidance capability and path optimization proficiency. Gray squares represent obstacles that are static and unknown, while yellow squares indicate dynamic obstacles. The blue triangle represents the robot's starting point, the blue circle represents the robot's endpoint, and the red * represents the local goal points. After conducting numerous tests, the robot's parameters are summarized in Table 2 below.

Table 2. Robot motion parameters.

Maximum line velocity	1 m/s	Maximum angular velocity	20 °/s
Maximum linear acceleration	0.2 m/s^2	Maximum angular acceleration	50 °/s
Linear velocity resolution	0.02 m/s	Angular velocity resolution	1 °/s

The results of the simulation experiment are presented in Figure 10. The black grid denotes the static obstacles that were previously identified, the yellow grid indicates the dynamic obstacles, and the gray grid represents the unknown static obstacles, mimicking the circumstance where unknown obstacles emerge on a known map within a chemical laboratory. When t was the specified time, the trajectories of the robot and the dynamic obstacle converged at point (6, 7) and drew closer to each other. Figure 10a reveals that the

traditional fusion algorithm strayed from the initial route to evade the dynamic obstacle but was trapped in a local optimum and failed to reach the destination. Figure 10b demonstrates the algorithm proposed in this work. The final statistical results are shown in Table 3. Through the adoption of the stop-and-wait strategy, replanning strategy, and adaptive heading strategy, it reached the destination safely.



(**a**) (**b**)

Figure 10. Comparative analysis of fusion algorithms in a dynamic environment. (**a**) The fusion algorithm presented in [25]. (**b**) The integrated algorithm presented in this article.

Table 3. The results obtained from the fusion algorithm.

Algorithm	Path Length (m)	Time (s)
Algorithm from [25]	——	——
Algorithm in this paper	30.37	146

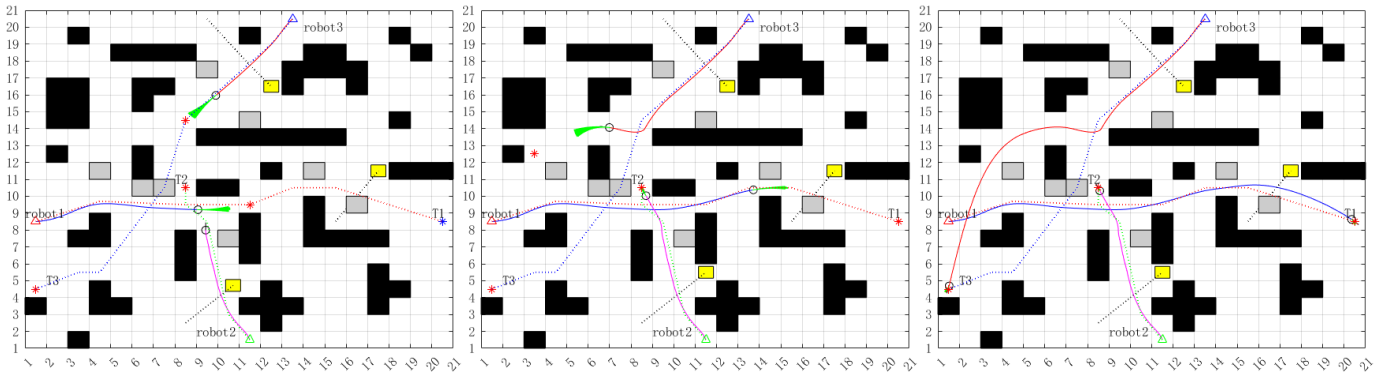
6.3. Simulation Experiment on Multi-Robot Coordination

To verify the efficacy of the multi-robot cooperative planning algorithm presented in this paper within complex circumstances, taking into account the time expenditure, this study carried out a simulation experiment on a 20 × 20 grid map. The experiment mimicked a situation where three robots performed their individual tasks in the same environment. The experimental setting encompassed dynamic obstacles and unknown static ones to heighten the complexity and difficulty of path planning. Each robot was allocated an independent starting point and destination, and the path needed to be adjusted dynamically during task execution to prevent conflicts with other robots and obstacles. The yellow grid indicates dynamic obstacles, the gray grid represents unknown static obstacles, and the black grid stands for known static obstacles. The simulation outcomes are presented in Figure 11, where the triangle indicates the starting point of the robot, and T1, T2, and T3 respectively denote the target points of Robot 1, Robot 2, and Robot 3.

At t = t0, a node conflict emerged between Robot 1 and Robot 2. Since Robot 1 had already passed through the intersection where the goal point of Robot 2 was located, Robot 1 was automatically assigned a higher priority. Moreover, as Robot 2 had no alternative route to bypass Robot 1 in the narrow section, Robot 2 stopped and waited. Robot 1 proceeded along its original path. Once it was safe, Robot 2 resumed its forward movement.

At t = t1, Robot 3 detected the existence of unknown static obstacles on the original path, while Robot 2 was on the verge of reaching the target point, thus Robot 2 was automatically assigned a higher priority. At this point, Robot 3 assessed the feasibility of the current path ahead and determined that proceeding further would lead to an inability

to bypass the obstacle or collide with Robot 2, resulting in a path deadlock. Consequently, Robot 3 chose to re-plan the path, and Robot 2 continued its forward movement.

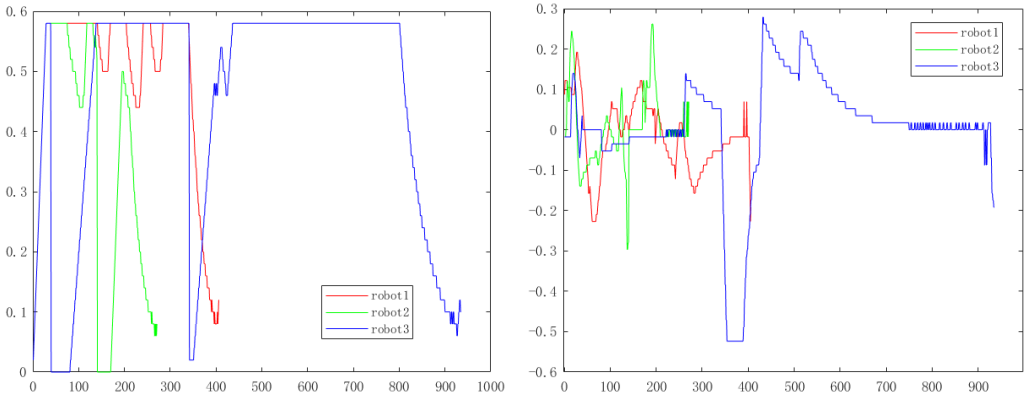


(**a**) t = t0 (**b**) t = t1 (**c**) t = t2

Figure 11. Multi-robot path planning.

At t = t2, the three robots managed to evade obstacles successfully and reached their respective target points concurrently via coordinated path planning. This process showcases the ability of multi-robot systems to collaborate effectively in complex environments.

As depicted in Figure 12a,b, the curves of the velocity and angular velocity of the robots are presented. When Robot 3 encountered a moving obstacle for the first time, it opted to halt and wait since the obstacle met the distance threshold and angle threshold; thus, its linear velocity was zero. Once the obstacle passed, Robot 3 resumed its movement. Due to the fact that the priority of Robot 2 was higher than that of Robot 3, Robot 3 undertook path replanning, and its linear velocity tended to zero. Robot 2 had a linear velocity of zero because it needed to avoid Robot 1. In conclusion, this validates that the algorithm proposed in this paper can effectively address the conflict issue between robots. The algorithm is capable of generating global paths efficiently in complex environments and flexibly circumventing dynamic and unknown static obstacles through local path planning to ensure the safe operation of the robots. The actual trajectories of the multiple robots in the complex dynamic environment demonstrate the effectiveness and viability of the proposed algorithm in optimizing paths, improving obstacle avoidance, and facilitating cooperation in multi-robot systems.



(**a**) Robot linear speed (**b**) Robot angular velocity

Figure 12. Robot parameters.

7. Discussion

To address the issue of path conflicts in multi-robot path planning within complex dynamic environments. This study introduces an enhanced A* algorithm integrated with the dynamic window approach for multi-robot path planning, specifically designed to address the challenges of path planning and conflict resolution for robots transporting reagents in chemical laboratories. Through enhancements to the traditional A* algorithm, the search time and redundant nodes were decreased, thereby enhancing the efficiency of global path planning. The dynamic window approach (DWA) enhances the robot's real-time obstacle avoidance capability by incorporating an adaptive heading angle strategy. This strategy allows the robot to flexibly adjust its heading based on dynamically sensed obstacles, effectively avoiding collisions and ensuring the safety and efficiency of path planning. In multi-robot path planning, the addition of a stop-and-wait mechanism can effectively resolve path conflicts, ensuring coordinated operation in a shared environment. Furthermore, by introducing a path re-planning strategy, the robot can promptly adjust its path when blocked by unknown static obstacles or other robots, preventing it from getting stuck and ensuring the smooth execution of tasks. The combination of these mechanisms makes the system more adaptable and robust in complex environments.

In the following stage of this work, we will deploy more robots in this environment and test the algorithm on a robotic platform to validate its performance in real-world scenarios.

Author Contributions: Y.H. and Z.A. conceived and designed the research; Y.H., Z.A., and C.L. carried out the experiment; Y.H. and C.L. wrote the manuscript. All the authors discussed the data and revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: Xinjiang Uygur Autonomous Region Central Government's Guidance Fund for Local Science and Technology Development Project (project number: ZYYD2025QY17).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Heselden, J.R.; Das, G.P. Heuristics and Rescheduling in Prioritised Multi-Robot Path Planning: A Literature Review. *Machines* **2023**, *11*, 1033. [CrossRef]
2. Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path planning techniques for mobile robots: Review and prospect. *Expert Syst. Appl.* **2023**, *227*. [CrossRef]
3. Qin, H.W.; Shao, S.L.; Wang, T.; Yu, X.T.; Jiang, Y.; Cao, Z.H. Review of Autonomous Path Planning Algorithms for Mobile Robots. *Drones* **2023**, *7*, 211. [CrossRef]
4. Luo, M.; Hou, X.R.; Yang, J. Surface Optimal Path Planning Using an Extended Dijkstra Algorithm. *IEEE Access* **2020**, *8*, 147827–147838. [CrossRef]
5. Duan, H.; Wu, Y.; Liu, J. Path planning of reconfigurable robot based on improved A* algorithm. *Elec. Meas. Tech.* **2024**, *46*, 44–50.
6. Li, Z.; Wu, J.; Wang, S.; Li, Y.; Yan, X.; Cheng, S.; Liu, G. Intelligent Arrangement of Ship Pipeline Based on Astar-Genetic Algorithm. *J. Xi'an Jiaotong Univ.* **2023**, *57*, 172–180.
7. Li, Y.B.; Soleimani, H.; Zohal, M. An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *J. Clean. Prod.* **2019**, *227*, 1161–1172. [CrossRef]
8. Luo, Q.; Wang, H.B.; Zheng, Y.; He, J.C. Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Comput. Appl.* **2020**, *32*, 1555–1566. [CrossRef]
9. Sun, Y.C.; Zhao, X.L.; Yu, Y.Z. Research on a Random Route-Planning Method Based on the Fusion of the A* Algorithm and Dynamic Window Method. *Electronics* **2022**, *11*, 2683. [CrossRef]
10. Lee, D.H.; Lee, S.S.; Ahn, C.K.; Shi, P.; Lim, C.C. Finite Distribution Estimation-Based Dynamic Window Approach to Reliable Obstacle Avoidance of Mobile Robot. *IEEE Trans. Ind. Electron.* **2021**, *68*, 9998–10006. [CrossRef]

11. Tong, X.L.; Yu, S.E.; Liu, G.Y.; Niu, X.D.; Xia, C.J.; Chen, J.K.; Yang, Z.; Sun, Y.Y. A hybrid formation path planning based on A* and multi-target improved artificial potential field algorithm in the 2D random environments. *Adv. Eng. Inform.* **2022**, *54*, 14. [CrossRef]
12. He, Z.B.; Chu, X.M.; Liu, C.G.; Wu, W.X. A novel model predictive artificial potential field based ship motion planning method considering COLREGs for complex encounter scenarios. *ISA Trans.* **2023**, *134*, 58–73. [CrossRef] [PubMed]
13. Wu, J.F.; Ma, X.H.; Peng, T.R.; Wang, H.J. An Improved Timed Elastic Band (TEB) Algorithm of Autonomous Ground Vehicle (AGV) in Complex Environment. *Sensors* **2021**, *21*, 8312. [CrossRef] [PubMed]
14. Zhu, H.; Jin, K.; Gao, R.; Wang, J.; Shi, C.J.R. Timed-Elastic-Band Based Variable Splitting for Autonomous Trajectory Planning. *arXiv* **2024**.
15. Yan, J.; Bei, J.; Yu, B.; Wu, W.; Xia, Z.; Tian, S.; Li, D. Indoor path planning based on improved A* algorithm. *Sci. Surv. Mapp.* **2023**, *48*, 58–67.
16. Dong, Y.; Yang, J.; Liu, W.; Zhang, B. Improved A* algorithm for robot full coverage connection path planning. *Transducer Microsyst. Tech.* **2023**, *42*, 125–128.
17. Chi, X.; Li, H.; Fei, J. Research on robot random obstacle avoidance method based on fusion of improved A* algorithm and dynamic window method. *Chin. J. Sci. Instrum.* **2021**, *42*, 132–140.
18. Abubakr, O.A.; Jaradat, M.A.; Abdel-Hafez, M.F. Intelligent Optimization of Adaptive Dynamic Window Approach for Mobile Robot Motion Control Using Fuzzy Logic. *IEEE Access* **2022**, *10*, 119368–119378. [CrossRef]
19. Yang, L.; Li, C.H.; Wang, Q.T.; Shi, S.X.; Yang, M.R. Adaptive dynamic windowing approach based on risk degree function. *Trans. Inst. Meas. Control* **2024**, *46*, 1542–1550. [CrossRef]
20. Zhang, C.; Li, Y.B.; Zhou, L.L. Optimal Path and Timetable Planning Method for Multi-Robot Optimal Trajectory. *IEEE Robot. Autom. Lett.* **2022**, *7*, 8130–8137. [CrossRef]
21. Yao, D.; Yin, X.; Luo, Z.; Wen, R.; Cheng, Z.; Zou, H. AGVS Path Planning Agorithm in Complex Environments. *J. South China Univ. Tech.* **2023**, *51*, 56.
22. Chen, G.; Yu, S. Improved A* Multi-Robot Bilevel Path Planning Algorithm. *Comput. Eng. Appl.* **2023**, *59*, 312–319.
23. Zhong, X.; Tian, J.; Hu, H.; Peng, X. Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. *J. Intell. Robot. Syst.* **2020**, *99*, 65–77. [CrossRef]
24. Yang, Z.; Li, J.L.; Yang, L.W.; Wang, Q.; Li, P.; Xia, G.F. Path planning and collision avoidance methods for distributed multi-robot systems in complex dynamic environments. *Math. Biosci. Eng.* **2023**, *20*, 145–178. [CrossRef]
25. Yuan, Q.; Wei, G.; Tian, X.; Shen, S. Mobile Robot Navigation Method Based on Fusion of Improved A* Algorithm and Dynamic Window Approach. *J. Chin. Comput. Syst.* **2023**, *44*, 334–339.