Study Protocol

Research on AGV Path Planning Integrating an Improved A* Algorithm and DWA Algorithm

Wenpeng Sang [1,2], Yaoshun Yue [1,2], Kaiwei Zhai [1,2] and Maohai Lin [1,2,*]

[1] State Key Laboratory of Biobased Material and Green Papermaking, Faculty of Light Industry, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, China; 10431221361@stu.qlu.edu.cn (W.S.); 10431221360@stu.qlu.edu.cn (Y.Y.); 10431221356@stu.qlu.edu.cn (K.Z.)

[2] Faculty of Light Industry, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, China

* Correspondence: mhlin@qlu.edu.cn

Abstract: With the rapid development of the economy and the continuous improvement of people's living standards, the printing and packaging industry plays an increasingly important role in people's lives. The traditional printing industry is a discrete manufacturing industry, relying on a large amount of manpower and manual operation, low production efficiency, higher labor costs, wasting of resources, and other issues, so the realization of printing factory intelligence to improve the competitiveness of the industry is an important initiative. Automatic guided vehicles (AGVs) are an important part of an intelligent factory, serving the function of automatic transportation of materials and products. To optimize the movement paths of AGVs, enhance safety, and improve transportation efficiency and productivity, this paper proposes an alternative implementation of the A* algorithm. The proposed algorithm improves search efficiency and path smoothness by incorporating the grid obstacle rate and enhancing the heuristic function within the A* algorithm's evaluation function. This introduces the evaluation subfunction of the nearest distance between the AGV, the known obstacle, and the unknown obstacle in the global path in the dynamic window approach (DWA algorithm), and reduces the interference of obstacles with the AGV in global path planning. Finally, the two improved algorithms are combined into a new fusion algorithm. The experimental results show that the search efficiency of the fusion algorithm significantly improved and the transportation time shortened. The path smoothness significantly improved, and the closest distance to obstacles increased, reducing the risk of collision. It can thus effectively improve the productivity of an intelligent printing factory and enhance its flexibility.

Keywords: printing plant; AGV; route planning; fusion algorithm

check for updates

Citation: Sang, W.; Yue, Y.; Zhai, K.; Lin, M. Research on AGV Path Planning Integrating an Improved A* Algorithm and DWA Algorithm. *Appl. Sci.* **2024**, *14*, 7551. https://doi.org/10.3390/app14177551

Academic Editor: Paolo Renna

Received: 1 July 2024
Revised: 22 August 2024
Accepted: 23 August 2024
Published: 27 August 2024

1. Introduction

In the printing industry, traditional production methods rely heavily on manual labor, resulting in low production efficiency and significant resource waste. The emergence of intelligent printing factories, where automated guided vehicles (AGVs) act as crucial "bridges," has optimized material and product transportation, reduced manual handling costs, improved production efficiency and accuracy, and provided vital support for intelligent operations. Path planning is a core function of AGVs, determining their routes, obstacle avoidance strategies, and task execution sequences, which directly impact transportation efficiency, safety, and production smoothness [1]. Path planning for mobile robots in warehouses or factories is divided into global [2] and local planning [3]. Global path planning involves algorithms such as the A* [4] (grid-based), ant colony [5] (bionics-based), and RRT [6] (random sampling-based), focusing on the overall route from start to end [7]. Local path planning, using methods like the dynamic window approach [8] and artificial potential field [9,10], addresses real-time adjustments and obstacle avoidance during travel. Most researchers combine global and local path planning to enable AGVs to synchronize dynamic adjustment and obstacle avoidance capabilities while planning the globally optimal path [11]. For example, Mohanraj et al. proposed a path tracking strategy based on matrices and feasible nodes to enhance a robot's computational ability and reaction speed during path planning, thereby reducing the chance of collisions between the mobile robot and obstacles. However, when more obstacles are present, multiple zero-potential energy points can cause a mobile robot to fall into a local optimal state [12]. Xinpeng Zhai et al. introduced a safe distance matrix to improve the illumination function and ensure safety during path searches. They also proposed an adaptive weight adjustment strategy to reduce redundant nodes and constructed a global evaluation function for dynamic real-time obstacle avoidance, although they did not smooth the path points, resulting in poor path smoothness [13]. Sunil Kumar et al. used a neighborhood search strategy based on reward value and probability to enrich and expand the search path. They also applied a path smoothing principle based on quasi-swing curves to enhance the path smoothness. However, they did not consider the safety distance, causing the globally planned paths to traverse obstacle prongs [14].

An automated guided vehicle (AGV) in a smart printing factory is a type of mobile robot used for automated transportation, handling, and coordination of production processes [15]. The complex environment of a printing factory—characterized by staff movement, intricate terrain, and densely packed equipment—increases the difficulty of path planning, thereby reducing the efficiency and accuracy of path searching. Additionally, the movement of staff and the dense arrangement of equipment significantly heighten the risk of collisions for the mobile robot. Therefore, an AGV in an intelligent printing factory must continuously sense these situations in real-time and make timely decisions, making path planning more complex.

This paper proposes an alternative implementation of the A* algorithm and a path planning method that integrates the improved dynamic window approach (DWA). The method involves analyzing the environment of a printing factory by constructing a map model, quantifying the obstacle rate, and incorporating this information into the evaluation function of the A* algorithm to enhance path search efficiency. The selection process for sub-nodes is optimized to prevent the mobile robot from traversing obstacle vertices, significantly reducing the collision rate. Additionally, the smoothing of path nodes is refined, allowing path node selection to move beyond the grid center, which decreases the number of transitions, shortens the path length, and improves path smoothness. An evaluation sub-function, which considers the nearest distance between the automated guided vehicle (AGV) and obstacles on the global path, and is incorporated into the dynamic window method to mitigate interference from both known and unknown obstacles. Ultimately, a global evaluation function integrates these two improved algorithms, enabling the AGV to perform dynamic real-time obstacle avoidance while following the global optimal path within the printing factory.

2. Alternative Implementation of the A* Algorithm

In this paper, we initially outline the fundamental principles of the A* algorithm and scrutinize the shortcomings of the traditional approach. Subsequently, we provide a detailed description of the grid obstacle rate and the optimized sub-node selection method.

2.1. Traditional A Algorithm*

The A* algorithm is a path planning algorithm that combines a heuristic function with a cost function, making it the most effective direct search method for solving the shortest path in static environments. It is also a highly effective algorithm for solving various search problems and is widely used in applications such as indoor robot pathfinding and game animation pathfinding [16]. The A* algorithm constructs a cost function based on the heuristic function, which considers both the cost of the distance from the new node to the initial point and the cost of the distance from the new node to the target point. The evaluation function of the A* algorithm is expressed as

$$F(n) = G(n) + H(n) \tag{1}$$

where $F(n)$ represents the estimated cost from the initial node through node n to the target node, $G(n)$ represents the actual cost from the initial node to node n, and $H(n)$ represents the estimated cost from node n to the target node. $H(n)$ is also known as the heuristic function of the A* algorithm, which controls the speed and accuracy of the algorithm. In most cases, the Euclidean distance [17] is used as a heuristic function for the A* algorithm [18]. The distance formula is

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{2}$$

where (X_1, Y_1) are the initial node coordinates and (X_2, Y_2) are the target node coordinates.

The A* algorithm utilizes an open list and a closed list to manage nodes to be expanded and those that have already been expanded, respectively [19]. When a node is searched, the adjacent nodes are added to the open list and their evaluation values are calculated using Equations (1) and (2). The node with the smallest evaluation value is then selected as the next node to be expanded and is added to the closed list. This process continues iteratively until the target node is added to the closed list, indicating that the A* algorithm has successfully found a path.

The path generated by the traditional A* algorithm consists of grid centroids, which can lead to unnecessary twists and turns [20] resulting in an unsmooth movement path. Additionally, the lines connecting the grid centroids often become tangential to obstacle vertices, posing a collision risk for mobile robots. To address these issues, this paper introduces the grid obstacle rate and optimized sub-node selection to the traditional A* algorithm.

2.2. Alternative A* Algorithm

In a grid map, obstacles are represented as unit grids, with their locations defined as obstacle grids. The more obstacle grids present, the more complex the path selection within the map becomes, resulting in the minimum path distance between the initial and target positions being greater than the Euclidean distance between the two points [21]. To estimate the minimum path distance, we introduce the obstacle rate Q to represent the environmental complexity within the map. The obstacle rate Q is defined as the ratio of the number of obstacle grids to the total number of grids within the local grid map between the current position and the target position. The total number of obstacle grids between the current and target positions is N. The coordinates of the initial position are (X_i, Y_i), and the coordinates of the target position are (X_t, Y_t). The expression is as follows:

$$Q = \frac{N}{(|X_i - X_t| + 1)(|Y_i - Y_t| + 1)} \ (Q \in (0, 1)) \tag{3}$$

In this formula, N represents the total number of grids within the matrix region defined by the initial and target positions.

In the evaluation function, the heuristic function $H(n)$ represents the estimated cost from the current position (X_c, Y_c) to the target position (X_t, Y_t). Specifically, $H(n)$ is the Euclidean distance from node n to the target position, which is the shortest path length between two points.

$$H(n) = \sqrt{(X_c - X_t)^2 + (Y_c - Y_t)^2} \tag{4}$$

In the presence of obstacles, the actual distance from node n to the target can only be equal to or greater than the Euclidean distance. Assuming $H(n)$ is greater than the actual minimum cost from node n to the target, denoted as $H^*(n)$, implies the existence of a node n where $H(n) > h^*(n)$. However, since the Euclidean distance $\sqrt{(X_n - X_t)^2 + (Y_n - Y_t)^2}$ is the shortest straight-line distance between two points, this situation cannot occur. Therefore, the assumption is invalid, meaning $H(n) \leq H^*(n)$, and the heuristic function $H(n)$ is admissible.

The evaluation function $F(n)$ consists of the actual cost $G(n)$ and the heuristic function $H(n)$, and adjusts the weight of the heuristic function based on the obstacle rate Q.

$$F(n) = G(n) + H(n)\left(e^{-lnQ}\right) \tag{5}$$

$$G(n) = \sum_{k=1}^{n} \sqrt{(X_k - X_{k-1})^2 + (Y_k - Y_{k-1})^2}(X_0, Y_0) = (X_i, Y_i) \tag{6}$$

Here, $G(n)$ is the actual cumulative cost from the starting position (X_i, Y_i) to the current position (X_c, Y_c).

In the evaluation function of the A* algorithm, the heuristic function $H(n)$ determines the search performance of the algorithm. The heuristic function values vary with the number of obstacles in different map environments. When there are fewer obstacles in the map, the obstacle rate Q is lower, causing the A* algorithm to be biased towards the target position during the search. This increases the heuristic function value $H(n)$, thus enhancing the search speed and directional accuracy towards the target position. Conversely, when the map contains many obstacles, the search speed decreases. Simply increasing the heuristic function value $H(n)$ to speed up the search can lead the AGV into local optima [22], resulting in more convoluted paths and higher distance costs. This forces the AGV to search multiple paths even in traversable areas, making the optimal path more difficult to find. Therefore, when there are many obstacles, the heuristic function value $H(n)$ should be reduced to lower the search speed and improve search precision, thereby obtaining the optimal global path.

As shown in Equation (3), the obstacle rate Q is jointly determined by the number of grids N, the initial position (X_i, Y_i), and the target position (X_t, Y_t). The search space of the A* algorithm changes with variations in the initial and target positions. After introducing the obstacle rate Q, the new heuristic function is defined as $H'(n) = H(n)\left(e^{-lnQ}\right)$. To ensure its admissibility, we must adjust this heuristic function. Since Q is a value between 0 and 1, and $e^{-lnQ} > 1$, the value of $H(n)$ is scaled up. Directly scaling $H(n)$ may violate the admissibility condition, so a dynamic adjustment parameter α is introduced where α = min (1, e^{-lnQ}). The new heuristic function is then defined as $H'(n) = \alpha \cdot H(n)$. When Q is large, indicating a complex map environment and difficult path search, the heuristic function $H(n)$ should be reduced to increase the search accuracy so α = 1 and $H'(n)$ = $H(n)$, maintaining admissibility. When Q is small, indicating a simple map environment and an easier path search, $H(n)$ should be increased to reduce the search space and speed up the process, so α = e^{-lnQ}. To ensure that $\alpha \cdot H(n)$ does not exceed the actual minimum cost $H^*(n)$, α = min (1, $\frac{H'(n)}{H(n)}$). With this definition, the new heuristic function $H'(n) = H(n)\left(e^{-lnQ}\right)$ satisfies $H'(n) \leq H^*(n)$ in all cases, ensuring the heuristic function's admissibility. In different regions, the search direction and speed are dynamically adjusted by modifying the weight of the heuristic function $H(n)$, allowing the A* algorithm to adapt during the search process $F(n)$. This approach achieves both the speed and flexibility of path planning, ensuring the planned path is globally optimal.

As shown in Equation (5), when there are fewer obstacles, the obstacle rate Q decreases which then increases the heuristic function value, reduces the search space during the A* algorithm's operation, and increases the path search speed. Conversely, when there are more obstacles, the obstacle rate Q increases, which reduces the weight of the heuristic function $H(n)$. This leads to an increase in path nodes during the A* algorithm's search process, thereby improving path search accuracy. Therefore, enhancing the evaluation function of the A* algorithm can improve its speed, accuracy, and adaptability to environmental information when searching different locations on the map.

2.3. Optimize the Way Child Nodes Are Selected

The storage space of the A* algorithm includes all current points in the path planning process and the surrounding child nodes [23]. The point with the lowest evaluation function cost is selected as the next path node. The selection of child nodes involves determining whether a location is an obstacle grid; if it is, no child node can be generated at that location. Therefore, AGVs that do not account for obstacle grids are prone to tangential contact with obstacle vertices or even collisions with the intersecting vertices of two obstacles during task execution. Figure 1 illustrates an AGV colliding with obstacle vertices during movement.

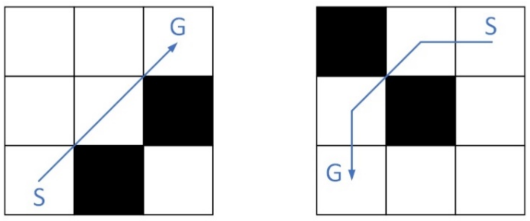

Figure 1. AGV movement path in A* algorithm.

To address the issue of robots easily colliding tangentially with obstacle vertices during the path search process of the traditional A* algorithm, this paper optimizes the selection method for child nodes. For obstacles encountered in the path planning process, this paper removes the child nodes around the obstacle nodes, preventing their selection. As shown in Figure 2, node n is the parent node, and nodes n_1 to n_8 are the child nodes. The optimized child node selection method is as follows:

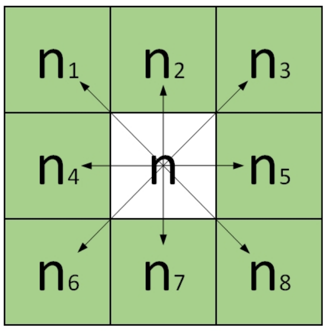

Figure 2. Node diagram.

a. When the obstacle node is n_2 or n_7, the left and right child nodes of the obstacle node are removed, i.e., nodes n_1 and n_3 or nodes n_6 and n_8;

b. When the obstacle node is n_4 or n_5, the upper and lower child nodes of the obstacle node are removed, i.e., nodes n_1 and n_6 or nodes n_3 and n_8;

c. When the obstacle node is n_1, n_3, n_6, or n_8, the child nodes are not processed;

By applying the above child node selection method, the A* algorithm can effectively prevent an AGV from cutting diagonally over obstacle vertices during the path planning process, reducing the risk of collisions between the AGV and obstacles in the printing factory. Figure 3 shows the optimized AGV movement path.

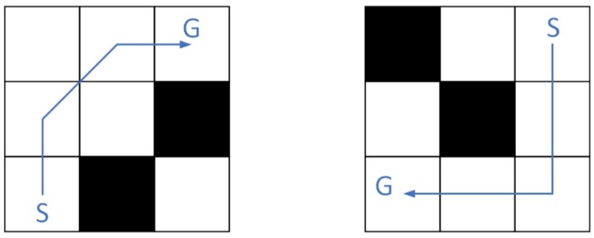

Figure 3. AGV movement path in optimized A* algorithm.

2.4. Smoothness Optimization

Since the A* algorithm's selection of child nodes in the path planning process is based on the current node, the generated path will contain many redundant nodes [24] and inflection points. Additionally, because the map is constructed using the raster method [25], the planned path comprises multiple sets of raster centers, resulting in a high number of transitions, as the path nodes can only be located at the center of the raster cells. This leads to a path that is neither the shortest nor smoothest. To address these issues, this paper proposes shortening the robot's motion path and improving the path smoothness by reducing the number of turning points. For two non-adjacent path nodes, if the connecting distance between them is less than the planned path distance, and the connecting line does not intersect obstacle vertices or pass through obstacles, the redundant nodes in the middle can be deleted. As shown in Figure 4, a_1 to a_3 represent three different paths. If a_1 and a_2 are co-located, then node B can be directly skipped to obtain path a_3. If a_1 and a_2 are not co-located, it is necessary to determine whether path a_3 intersects obstacle vertices or passes through an obstacle. If a_3 intersects obstacle vertices or passes through an obstacle, the original path is retained. If not, the new path a_3 is obtained and the original paths a_1 and a_2 are eliminated.

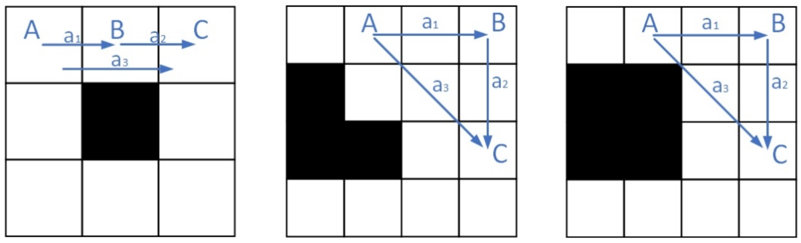

Figure 4. Path turn optimization.

Take Figure 5 as an example; S is the initial point of the path, G is the target point, and the motion path before optimization is S-n_1-B-n_2-G. The entire path contains redundant nodes and many turns. Using the method proposed in this paper, the path is optimized for smoothness. First, intermediate points on the same connecting line are deleted, retaining only the starting point, inflection points, and end point, resulting in the path S-A-B-G. Next, the initial point S is connected to path node B to form a new path b. If path b passes through an obstacle, it is discarded; otherwise, the path is updated to S-B-G. Finally, the midpoint C of path b is selected and connected to the goal point G to form the new path c. If path c passes through an obstacle, it is discarded; otherwise, the optimized path S-C-G is obtained.

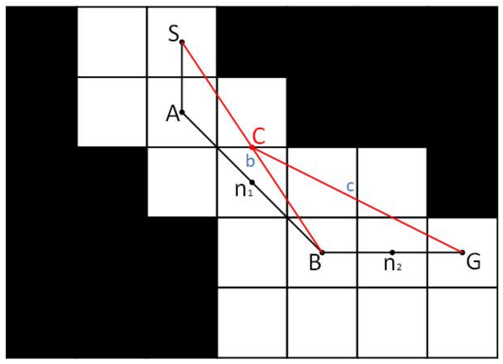

Figure 5. Turn-optimized movement paths.

To verify the performance of the alternative A* algorithm, this paper conducts a comparative experiment using the same starting and ending points. Simulations are performed with both the traditional A* algorithm and the alternative A* algorithm, and the experimental results are presented in Figure 6. The grid map size is 20 × 20, and the length of each grid unit is 1 m. As shown in Figure 6a, the path planned by the traditional A* algorithm is closer to the obstacles, which can easily lead to collisions between the AGV and the obstacles, resulting in poor safety of the global path. Figure 6b demonstrates that after adopting the obstacle grid method and optimizing the child node selection method, the distance between AGVs and obstacles in the global path increases significantly, avoiding collisions and improving the path safety. Compared with the traditional A* algorithm, the path distance planned by the algorithm with smoothness optimization is significantly shorter, and the AGV's movement process is smoother.

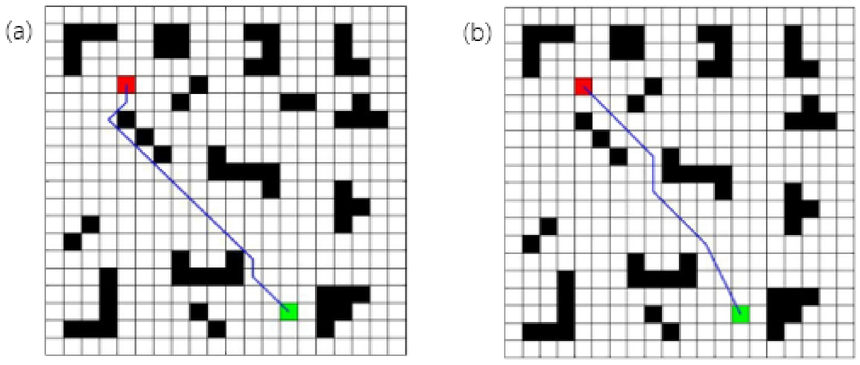

Figure 6. Conventional A* algorithm and improved A* algorithm path diagram, (**a**) xonventional A* algorithm path diagram, (**b**) alternative A* algorithm path diagram.

As shown in Table 1, the improved A* algorithm increases the distance to the nearest obstacle by 0.5 m compared to the alternative A* algorithm, thereby improving the path safety. Additionally, the total path length is shortened by 0.828 m, and the path planning time is reduced by 0.76 s, enhancing the path planning efficiency.

Table 1. Comparison results of traditional A* algorithm and alternative A* algorithm.

	Path Length (m)	**Path Search Time (s)**	**Distance from Path to the Nearest Obstacle (m)**
Traditional A* algorithm	17.556	16.11	0
Alternative A* algorithm	16.728	15.35	0.5

3. Improvement of the DWA Algorithm

Although the improved A* algorithm can obtain the global optimal path from the initial point to the target point, it does not account for unknown obstacles. In the complex road conditions of a printing factory, the AGV may collide with moving obstacles during travel. To address these issues, this paper proposes an improvement to the traditional dynamic window approach (DWA) algorithm.

3.1. Traditional DWA Algorithm

The traditional dynamic window approach (DWA) algorithm simulates the motion trajectory of a mobile robot by collecting its velocity samples and predicting its motion trajectory at the sampled velocities. Among all of the feasible trajectories predicted, the optimal trajectory is determined by evaluating the evaluation function [26]. In the trajectory prediction stage, the robot's kinematic model is used for trajectory prediction. Assuming that the mobile robot moves at a time interval Δt, the kinematic model of the mobile robot can be obtained from Equation (7). Here, (X_1, Y_1, θ_1) represents the robot's position and orientation at time t, and (X_2, Y_2, θ_2) represents its position and orientation at time $t + \Delta t$, with $\triangle\theta = \omega_t \Delta t$.

$$\begin{cases} X_2 = X_1 + V\_tcos(\triangle\theta)\Delta t \\ Y_2 = Y_1 + V\_tsin(\triangle\theta)\Delta t \\ \theta_2 = \theta_1 + \omega_t \Delta t \end{cases} \tag{7}$$

The motion state of the robot is jointly determined by its current linear velocity v and angular velocity ω. Under the velocity constraints of the robot, the sampling space of the velocity sets is established, resulting in multiple velocity sets (v_t, ω_t) in the velocity space. However, in practical applications, the sampling velocity range is limited by the constraints of the mobile robot's performance and the surrounding environment. The sampling velocity range of the mobile robot can be expressed as follows:

$$V_m = \{(v, m)|v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \tag{8}$$

where v_{min} and v_{max} are the minimum and maximum linear velocities of the mobile robot, respectively; ω_{min} and ω_{max} are the minimum and maximum angular velocities of the mobile robot, respectively.

Since the mobile robot is driven by a motor, the addition and subtraction of velocity is constrained by the motor. The sampled velocity space when acceleration is considered within the dynamic window can be expressed as

$$V_d = \left\{(v, \omega)\middle|v \in \left[v_t - \dot{v}\Delta t, v_t + \dot{v}\Delta t\right], \omega \in \left[\omega_t - \dot{\omega}\Delta t, \omega_t + \dot{\omega}\Delta t\right]\right\} \tag{9}$$

where (v_t, ω_t) is the current velocity group of the mobile robot, and $\dot{v}$ and $\dot{\omega}$ are the maximum linear acceleration and maximum angular acceleration of the mobile robot, respectively.

The mobile robot is also constrained by its braking distance. In the localized obstacle avoidance process, the safety of the mobile robot must be ensured, and the robot needs to brake to a stop before colliding with an obstacle. This means that under the condition of maximum deceleration, the current velocity can be reduced to zero before a collision occurs. The velocity limit of the robot can be expressed as follows:

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

where $dist(v, \omega)$ denotes the closest distance of the mobile robot to the obstacle at the current speed.

With the above constraints, the final velocity sampling space of the mobile robot is the intersection of the three velocity spaces, as shown in Equation (11):

$$\begin{gathered} V_s = V_m \cap V_d \cap V_a = \{(v, \omega)|v \in [v_{min}, v_{min}] \cap \omega \in [\omega_{min}, \omega_{min}]\} \\ V_s = V_m \cap V_d \cap V_a \end{gathered} \tag{11}$$

The predicted trajectory evaluation function of the DWA algorithm includes an azimuthal declination evaluation subfunction, a velocity evaluation subfunction, and an obstacle distance evaluation subfunction. This can be expressed as follows:

$$G(v, \omega) = \alpha Head(v, \omega) + \beta vel(v, \omega) + \gamma dist(v, \omega) \tag{12}$$

where $Head(v, \omega)$ is the evaluation subfunction of azimuthal declination, $vel(v, \omega)$ is the evaluation subfunction of the current velocity magnitude, and $dist(v, \omega)$ is the evaluation subfunction of the closest distance between the mobile robot and the obstacle at the corresponding velocity. The coefficients α, β, and γ are the weighting coefficients of the three evaluation subfunctions. Since the DWA algorithm prioritizes the safety of the mobile robot, it is defined as $\beta < \alpha < \gamma$.

3.2. Improvements to the DWA Algorithm

To ensure the safety of the mobile robot in the global path, the traditional DWA algorithm stipulates that the weight of the evaluation subfunction for the nearest distance between the robot and obstacles should always be larger than the weight of other subfunctions. However, in environments with more obstacles, this can lead to a local optimal state, increasing the robot's moving distance [27]. To address these problems, this paper designs an azimuth evaluation subfunction $Heading(v, \omega)$ that fuses the global path nodes, representing the azimuth deviation between the current path node and the predicted target node. Additionally, two evaluation subfunctions for the robot's distance to obstacles are designed: $dist\_sta(v, \omega)$ and $dist\_dyna(v, \omega)$. The $dist\_sta(v, \omega)$ subfunction evaluates the nearest distance between the mobile robot and the known obstacles in the global path nodes, while the $dist\_dyna(v, \omega)$ subfunction evaluates the nearest distance between the mobile robot and unknown obstacles in the local path nodes. The evaluation function can be expressed as follows:

$$\begin{gathered} G(v, \omega) = \alpha Heading(v, \omega) + \beta vel(v, \omega) + \\ \gamma dist\_sta(v, \omega) + \delta dist\_dyna(v, \omega) \end{gathered} \tag{13}$$

where α, β, γ and δ are the weighting coefficients of the four evaluation subfunctions, which are used to set different weights.

$Heading(v, \omega)$ represents the azimuthal deviation $\Delta\varphi$ of the predicted trajectory direction from the global path nodes. It updates the current goal point in the order of the global path nodes and updates in real-time when a new goal node is reached, facilitating the dynamic setting of the goal node. $dist\_sta(v, \omega)$ evaluates the closest distance of the mobile robot to known obstacles in the global path, reducing their interference in global path planning and avoiding the increased moving distances caused by local path planning away from obstacles. $dist\_dyna(v, \omega)$ evaluates the closest distance of the mobile robot to unknown obstacles in local path planning, enhancing the robot's ability to avoid unknown obstacles by increasing its weighting coefficients. After introducing the new subfunctions, by adjusting the weighting coefficients γ and δ, the interference of known obstacles on global path planning can be reduced, and the obstacle avoidance ability for unknown obstacles can be improved.

To verify the path planning effectiveness of the improved DWA algorithm, this paper conducts simulation experiments to compare and validate the results. The experiments use the same start and end points, setting the grid map size to 20×20 with a unit length of 1 m. The experimental results are shown in Figure 7. In Figure 7a, the traditional DWA algorithm prioritizes real-time and dynamic adaptation in the path planning process at the expense of the global optimum. This leads to the creation of local paths that deviate from the optimal paths, increasing the unnecessary path lengths in complex environments. Additionally, AGVs collide with obstacles during movement, reducing path safety. In Figure 7b, the improved DWA algorithm effectively addresses the local optimum issue and shortens the path length. The improved path length is 16.401 m compared to the 19.255 m path planned by the traditional DWA algorithm, resulting in a reduction of 2.854 m or 14.8%. The closest distance to obstacles is increased by 0.56 m, an improvement of 56%, enhancing the safety of the AGV during movement and reducing the risk of collisions with obstacles.

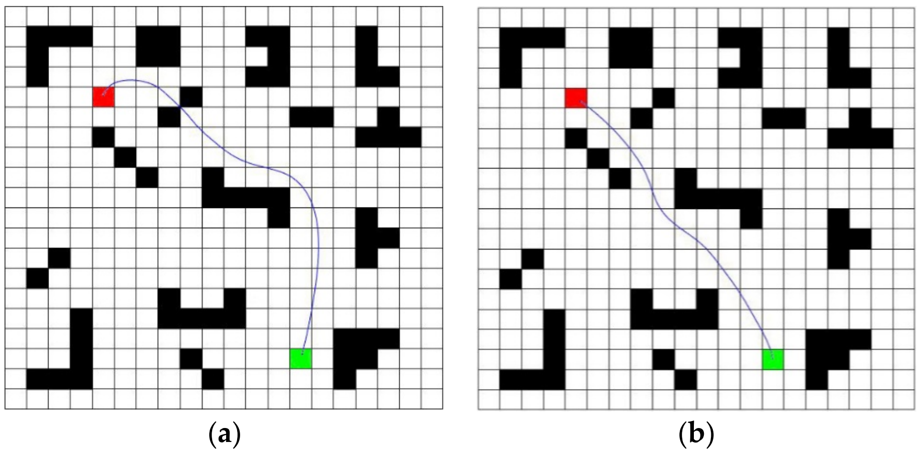

Figure 7. Path diagrams of traditional DWA algorithm and improved DWA algorithm. (**a**) Traditional DWA algorithm path. (**b**) Improved DWA algorithm path.

4. Hybrid Algorithm

In this paper, we propose an alternative implementation of the A* algorithm and optimize the DWA algorithm. While the proposed alternative A* algorithm enhances global path planning, it lacks the capability to avoid unknown and moving obstacles. On the other hand, the improved DWA algorithm can avoid unknown obstacles but the path it generates is not globally optimal. Therefore, this paper presents a fusion algorithm that combines the alternative A* algorithm with the DWA algorithm [28]. The flowchart of the fusion algorithm is shown in Figure 8.

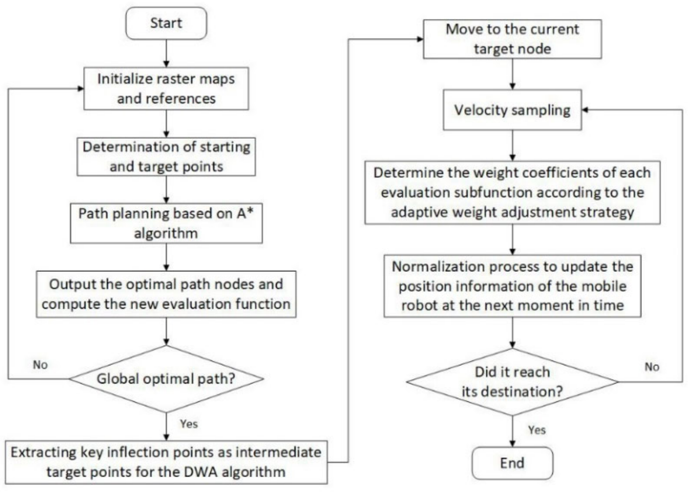
Start
Initialize raster maps and references
Determination of starting and target points
Path planning based on A* algorithm
Output the optimal path nodes and compute the new evaluation function
Global optimal path?
No
Yes
Extracting key inflection points as intermediate target points for the DWA algorithm
Move to the current target node
Velocity sampling
Determine the weight coefficients of each evaluation subfunction according to the adaptive weight adjustment strategy
Normalization process to update the position information of the mobile robot at the next moment in time
Did it reach its destination?
No
Yes
End

Figure 8. Flowchart of fusion algorithm.

As shown in Figure 8, global path planning is first performed by the A* algorithm to obtain the optimal global static path and extract the critical inflection points of the global path. Then, the critical inflection points, other than the starting point, are sequentially set as the local target points of the DWA algorithm, and real-time local planning is performed until the target point is reached. Combining the two algorithms not only plans the global optimal path and improves the smoothness of the path but also enhances the obstacle search efficiency and enables real-time dynamic obstacle avoidance.

5. Simulation and Experimental Analysis

To verify the feasibility, applicability, and reliability of the fusion algorithm proposed in this paper, simulation experiments were conducted to compare the alternative A* algorithm, the improved DWA algorithm, and the proposed fusion algorithm. MATLAB 2019b was used for programming. The raster map size was 20 $\times$ 20, with each small raster unit set to 1 m in length and black rasters indicating obstacles. A dynamic obstacle was randomly added to the simulation environment to verify the dynamic obstacle avoidance performance of the fusion algorithm. To ensure the fairness of the simulation experiments, the same map was used for all experiments, and the starting and target points for each algorithm's path planning were also the same. In the simulation experiment, the initial weights of each evaluation subfunction were set as follows: α = 0.2, β = 0.4, γ = 0.2, and δ = 0.35. The speed parameters are set as follows (Table 2).

Table 2. Speed parameter settings.

Parameter name	Numerical Value
Maximum line speed v_{max}	2 m/s
Maximum angular velocity ω_{max}	0.7 rad/s
Maximum linear acceleration $\dot{v}_{max}$	0.4 m/s^2
Maximum angular acceleration $\dot{\omega}_{max}$	1.7 rad/s^2
Projection time T	3 s

The simulation results of the three algorithms are shown in Figure 9. The green grid represents the starting point, the red grid represents the target point, the yellow square simulates the dynamic obstacles, and the gray square simulates the unknown static obstacles.

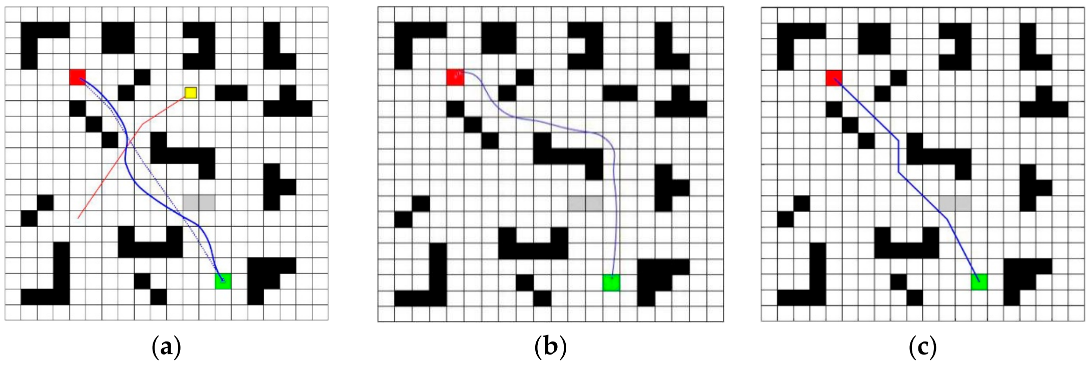

Figure 9. Simulation results of different algorithms. (**a**) Path planning of the fusion algorithm. (**b**) Path planning of the improved DWA algorithm. (**c**) Path planning with alternative A* algorithm.

As shown in Figure 9, while the alternative A* algorithm can find the globally optimal solution during path planning, the mobile robot may collide with obstacles when navigating around the unknown obstacles, as it lacks the ability to avoid them. The improved DWA algorithm is effective at avoiding both known and unknown static obstacles during path planning, preventing collisions. However, the path it plans in a complex global environment is not optimal, making the AGV's global path more costly. This paper addressed these issues by combining the two algorithms into a new fusion algorithm, enhancing the AGV's ability to avoid dynamic obstacles. As shown in Figure 9a, the fusion algorithm not only finds the global optimal solution in the global path planning process but also effectively avoids unknown static and dynamic obstacles, improving the path safety. Additionally, the robot's path was smoother during movement.

To further verify the effectiveness of the fusion algorithm, the path length, path search time, and the distance from the path to the nearest obstacle were used as performance metrics to quantitatively analyze the three algorithms mentioned above. The results are shown in Table 3.

Table 3. Results of the three algorithm runs.

	Path Length (m)	**Path Search Time (s)**	**Distance from Path to Nearest Obstacle (m)**
Alternative algorithm for A* algorithm	16.728	15.35	0
Improvement of the DWA algorithm	19.575	33.93	0.34
Fusion Algorithms	16.617	14.84	0.48

In this paper, the fusion algorithm reduces the path length by 0.111 m, shortens the path search time by 0.51 s compared to the improved DWA algorithm, and increases the shortest distance from the path to the obstacle by 0.14 m. The data in Table 3 demonstrate that the fusion algorithm performs well in global path planning, and its comprehensive performance is better than those of the other two algorithms.

6. Conclusions

This paper proposes an AGV path planning algorithm based on a grid obstacle rate and a sub-function weight adjustment strategy. This approach addresses the issues of low security in traditional A* algorithm path planning and the tendency for the DWA algorithm to converge on local optimal solutions. The A* algorithm plans a more secure and smoother global path by incorporating obstacle rate information and improving the heuristic function. The evaluation subfunction for the nearest distance between the mobile robot and known obstacles in the global path nodes, and the evaluation subfunction for the nearest distance between the mobile robot and unknown obstacles in the local path nodes, are introduced in the DWA algorithm. The weights of the subfunctions are adjusted to reduce the interference of known obstacles on global path planning and to improve the obstacle avoidance ability for unknown obstacles. This paper designs a new evaluation function to integrate the improved A* and DWA algorithms. In the fusion algorithm, the key inflection points generated by the alternative A* algorithm are sequentially set as local target points for the DWA algorithm, ensuring that the path planned by the DWA algorithm remains closer to the global path. The simulation and experimental results verify that the fusion algorithm proposed in this paper can safely and efficiently plan the global path, demonstrating its practical application value. Future research can further explore the AGV path planning problem in printing factories.

Funding: This research received no external funding.

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

References

1. Zhang, Z.; Wu, L.; Zhang, W.; Peng, T.; Zheng, J. Energy-efficient path planning for a single-load automated guided vehicle in a manufacturing workshop. *Comput. Ind. Eng.* **2021**, *158*, 107397. [CrossRef]
2. Warren, C.W. Global path planning using artificial potential fields. In Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; pp. 316–321.
3. Langer, D.; Rosenblatt, J.; Hebert, M. A behavior-based system for off-road navigation. *IEEE Trans. Robot. Autom.* **1994**, *10*, 776–783. [CrossRef]
4. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
5. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; pp. 1470–1477.
6. LaValle, S.M.; Kuffner, J.J. Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan. In *Algorithmic and Computational Robotics*; A K Peters/CRC Press: Natick, MA, USA, 2001; pp. 303–307. [CrossRef]
7. 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*, 120254. [CrossRef]
8. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
9. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [CrossRef]
10. Sun, Y.; Wang, W.; Xu, M.; Huang, L.; Shi, K.; Zou, C.; Chen, B. Local path planning for mobile robots based on fuzzy dynamic window algorithm. *Sensors* **2023**, *23*, 8260. [CrossRef]
11. Fragapane, G.; De Koster, R.; Sgarbossa, F.; Strandhagen, J.O. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *Eur. J. Oper. Res.* **2021**, *294*, 405–426. [CrossRef]
12. Mohanraj, T.; Dinesh, T.; Guruchandhramavli, B.; Sanjai, S.; Sheshadhri, B. Mobile robot path planning and obstacle avoidance using hybrid algorithm. *Int. J. Inf. Technol.* **2023**, *15*, 4481–4490. [CrossRef]
13. Zhai, X.; Tian, J.; Li, J. A Real-time Path Planning Algorithm for Mobile Robots Based on Safety Distance Matrix and Adaptive Weight Adjustment Strategy. *Int. J. Control Autom. Syst.* **2024**, *22*, 1385–1399. [CrossRef]
14. Kumar, S.; Sikander, A. An intelligent optimize path planner for efficient mobile robot path planning in a complex terrain. *Microsyst. Technol.* **2023**, *29*, 469–487. [CrossRef]
15. Vlachos, I.; Pascazzi, R.M.; Ntotis, M.; Spanaki, K.; Despoudi, S.; Repoussis, P. Smart and flexible manufacturing systems using Autonomous Guided Vehicles (AGVs) and the Internet of Things (IoT). *Int. J. Prod. Res.* **2024**, *62*, 5574–5595. [CrossRef]
16. Karur, K.; Sharma, N.; Dharmatti, C.; Siegel, J.E. A survey of path planning algorithms for mobile robots. *Vehicles* **2021**, *3*, 448–468. [CrossRef]
17. Laureijs, R.; Amiaux, J.; Arduini, S.; Augueres, J.-L.; Brinchmann, J.; Cole, R.; Cropper, M.; Dabin, C.; Duvet, L.; Ealet, A. Euclid definition study report. *arXiv* **2011**, arXiv:1110.3193.
18. Foead, D.; Ghifari, A.; Kusuma, M.B.; Hanafiah, N.; Gunawan, E. A systematic literature review of A* pathfinding. *Procedia Comput. Sci.* **2021**, *179*, 507–514. [CrossRef]

19. Yoshizumi, T.; Miura, T.; Ishida, T. A* with Partial Expansion for Large Branching Factor Problems. In Proceedings of the AAAI/IAAI, Austin, TX, USA, 30 July–3 August 2000; pp. 923–929.
20. Antikainen, H. Using the hierarchical pathfinding A* algorithm in GIS to find paths through rasters with nonuniform traversal cost. *ISPRS Int. J. Geo-Inf.* **2013**, *2*, 996–1014. [CrossRef]
21. Tsardoulias, E.G.; Iliakopoulou, A.; Kargakos, A.; Petrou, L. A review of global path planning methods for occupancy grid maps regardless of obstacle density. *J. Intell. Robot. Syst.* **2016**, *84*, 829–858. [CrossRef]
22. Baxter, J. Local optima avoidance in depot location. *J. Oper. Res. Soc.* **1981**, *32*, 815–819. [CrossRef]
23. Yao, J.; Lin, C.; Xie, X.; Wang, A.J.; Hung, C.-C. Path planning for virtual human motion using improved A* star algorithm. In Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 12–14 April 2010; pp. 1154–1158.
24. Chi, W.; Ding, Z.; Wang, J.; Chen, G.; Sun, L. A generalized Voronoi diagram-based efficient heuristic path planning method for RRTs in mobile robots. *IEEE Trans. Ind. Electron.* **2021**, *69*, 4926–4937. [CrossRef]
25. Delamater, P.L.; Messina, J.P.; Shortridge, A.M.; Grady, S.C. Measuring geographic access to health care: Raster and network-based methods. *Int. J. Health Geogr.* **2012**, *11*, 15. [CrossRef]
26. Chang, L.; Shan, L.; Jiang, C.; Dai, Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robot.* **2021**, *45*, 51–76. [CrossRef]
27. Mitsch, S.; Ghorbal, K.; Platzer, A. On provably safe obstacle avoidance for autonomous robotic ground vehicles. In Proceedings of the Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, 24–28 June 2013.
28. Ji, X.; Feng, S.; Han, Q.; Yin, H.; Yu, S. Improvement and fusion of A* algorithm and dynamic window approach considering complex environmental information. *Arab. J. Sci. Eng.* **2021**, *46*, 7445–7459. [CrossRef]