electronics

MDPI

Article

Path-Planning Strategy: Adaptive Ant Colony Optimization Combined with an Enhanced Dynamic Window Approach

Dongri Shan [1,2,*], Shuaishuai Zhang [3], Xiaofang Wang [3] and Peng Zhang [3]

[1] School of Mechanical Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, China
[2] School of Electronics and Information, Aerospace Information Technology University, Jinan 250200, China
[3] School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250300, China; 10431210532@stu.qlu.edu.cn (S.Z.); wxf2012@stu.xjtu.edu.cn (X.W.); zp@qlu.edu.cn (P.Z.)
* Correspondence: shandongri@qlu.edu.cn

Abstract: Aiming to resolve the problems of slow convergence speed and inability to plan in real time when ant colony optimization (ACO) performs global path planning, we propose a path-planning method that improves adaptive ant colony optimization (IAACO) with the dynamic window approach (DWA). Firstly, the heuristic information function is modified, and the adaptive adjustment factor is added to speed up the algorithm's convergence rate; secondly, elite ants and max–min ants systems are implemented to enhance the global pheromone updating process, and an adaptive pheromone volatilization factor is aimed at preventing the algorithm from enhancing its global search capabilities; then, the path optimization and withdrawal mechanism is utilized to enable smoother functioning and to avoid the deadlocks; finally, a new distance function is introduced in the evaluation function of DWA to the enhance real-time obstacle-avoidance ability. The simulation experiment results reveal that the path length of the IAACO can be shortened by 10.1% and 13.7% in contrast to the ACO. The iteration count can be decreased by 63.3% and 63.0%, respectively, leading to an enhanced optimization performance in global path planning and achieving dynamic real-time obstacle avoidance for local path planning.

Keywords: path planning; ant colony optimization; dynamic window approach; dynamic obstacle avoidance

check for updates

Citation: Shan, D.; Zhang, S.; Wang, X.; Zhang, P. Path-Planning Strategy: Adaptive Ant Colony Optimization Combined with an Enhanced Dynamic Window Approach. *Electronics* **2024**, *13*, 825. https://doi.org/10.3390/electronics13050825

Academic Editor: Andrea Bonci

Received: 20 January 2024
Revised: 15 February 2024
Accepted: 19 February 2024
Published: 20 February 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the continuous expansion of mobile robot application areas, the demand for efficient, flexible, intelligent path-planning algorithms is also gradually increasing. Path planning, as one of the core problems in mobile robotics, refers to the realization process of selecting an appropriate route to reach the goal point according to the constraints in a dynamic environment [1]. The commonly used traditional path-planning algorithms are ant colony optimization [2,3], genetic algorithm [4,5], fast expanded randomized trees [6,7], artificial potential field method [8], deep reinforcement learning [9,10], etc.

Ant colony optimization has good performance in the field of path planning. Still, there are problems, such as the tendency to fall into local optimums, slow convergence, and the inability to avoid obstacles in real time in global planning [11]. Some scholars optimize the performance by changing the traditional ant colony optimization or fusing other algorithms. Luo et al. [12] improves the ACO by introducing initial pheromones with unequal allocation, using pseudo-random state transition rules to select paths, and introducing optimal and worst solutions to improve the global pheromone update method, thereby enhancing the global optimal searchability and convergence speed. Tian et al. [13] have proposed an improved ant colony algorithm that uses a multi-step search strategy

instead of a single-step search strategy, redesigns the pheromone update mechanism, configures path smoothing, and improves algorithm performance and planning efficiency. Li et al. [14] propose an adaptive ant colony algorithm that optimizes the initial pheromone matrix using an adaptive heuristic function. Improved pheromone updates have improved the convergence speed and stability of planning paths. Chen et al. [15] establish an environmental model and a three-color grating image. Considering the length of the path and turning elements, optimizing the pheromone update mechanism, removing points, smoothing the planned path, and improving the algorithm can effectively design the optimal path. Si et al. [16] dynamically change the safety distance of obstacles and introduce a new evaluation term in the evaluation function. In a dynamic obstacle environment, it can ensure that the robot reaches the target point smoothly and shortens the time spent on the entire process. Wang et al. [17] have redesigned the kinematic model of the underwater vehicle and the evaluation function for local path planning based on DWA to ensure that the vehicle can adjust to the target point and avoid obstacles promptly during navigation. Jin et al. [18] propose a fusion algorithm RACO in complex dynamic environments, improves ant colony optimization, and proposes a two-level safety distance determination rule to handle unknown obstacles. It has good performance advantages in path planning and dynamic obstacle avoidance. Wang et al. [19] proposes a distributed obstacle-avoidance algorithm for multiple mobile robots to coordinate path planning and motion navigation between multiple robots and between robots and unknown areas. This algorithm combines ant colony optimization and dynamic window methods, coordinates multi-robot systems through priority policies, and can achieve collaborative obstacle avoidance of robots in unknown environments with high security and global optimality. Xiong et al. [20] propose a path-planning method that integrates improved ant colony and dynamic window algorithms. Introducing initial grid transfer rules, changing pheromone update methods, and introducing DWA for local path planning improves convergence speed, path smoothness, safety, and reliability and achieves dynamic obstacle avoidance of robots. Zhao et al. [21] proposes a smooth path-planning method that integrates an A * ant colony and dynamic window method, improves the A * algorithm by non-uniformly allocating initial pheromones, customizes movement step size and search method, adopts a quadratic path optimization strategy, and introduces evaluation sub-functions in the evaluation function of DWA to ensure the planning of the global optimal path and avoidance of dynamic obstacles.A summary of the previous research is provided in Table 1, including the objectives of each task, the methods used, and the results achieved.

Table 1. Summary table of previous research.

Predecessor Research	Target	Method	Result
[12–15]	Enhance the efficiency of global path planning	Improve ACO	Improved path-planning efficiency, but did not consider dynamic environments.
[16,17]	Path planning adapted to dynamic environments	DWA	Capable of handling dynamic environments without considering global optimal solutions
[18–21]	Realize global path-planning optimality and local dynamic path-planning accuracy	Combine ACO and DWA	Achieved good path-planning results in both static and dynamic environments

Although the above literature has improved path-planning performance and solved local obstacle-avoidance problems through algorithm fusion, there are still some shortcomings. In this article, we propose a method that combines adaptive ant colony optimization with enhanced dynamic windows to address the challenges of global and local path planning for mobile robots. The main contributions are as follows:

(1) Improvement for ant colony optimization: Adding adaptive adjustment factors based on heuristic information in traditional algorithms. Adjust the mechanism for updating ant pheromones by adding adaptive pheromone volatilization factors, elite ants, and maximum, minimum, and systems and using path optimization and exit mechanisms to avoid deadlocks.
(2) Use key points in global path planning as target points for local path guidance and real-time obstacle avoidance. A new distance function has been introduced into the evaluation function of DWA to enhance real-time obstacle-avoidance capability.
(3) Complete three sets of experiments: mainly focusing on the feasibility of adaptive ant colony optimization in different complex environments, the obstacle avoidance effect of fusion algorithms in different obstacle information, and verifying the algorithm's effectiveness in actual vehicle scenarios.

In summary, the content of this paper is as follows: Section 2 will introduce the basic theory of traditional ACO algorithms. Section 3 introduces our improved ACO algorithm. Section 4 describes the overall rules of the fusion algorithm. Section 5 discusses the experiments conducted. Section 6 summarizes the overall situation and proposes ideas for further research.

2. Introduction to the Basic Algorithm

2.1. Map Environment Modeling

The raster method is a widely employed technique for modeling environmental maps in path planning. It is adaptable to obstacles and can significantly reduce the complexity of modeling the environment [22]. It uses a combination of the right-angled coordinate system and sequence number for grid identification, as illustrated in Figure 1a. Among them, the white grid represents the free area, and the black grid represents the obstacle area. Equation (1) illustrates the transformation relationship between the grid sequence number and its corresponding coordinates. In Figure 1b, the grid i has eight selectable directions, and the following feasible raster selection is the raster formed in these eight directions of the adjacency matrix, where the red dots represent potentially selectable nodes.

$$\begin{cases} x_i = a(mod(i, N_x) - 0.5) \\ y_i = a(N_y + 0.5 - ceil(i/N_y)) \end{cases} \tag{1}$$

where (x_i, y_i) represents the positional coordinate of the ith grid; i denotes the sequence number; $mod()$ stands for the modulo operator; $ceil()$ represents the rounding operator towards positive infinity; N_x and N_y indicate the quantity of grids in both the row and column directions.

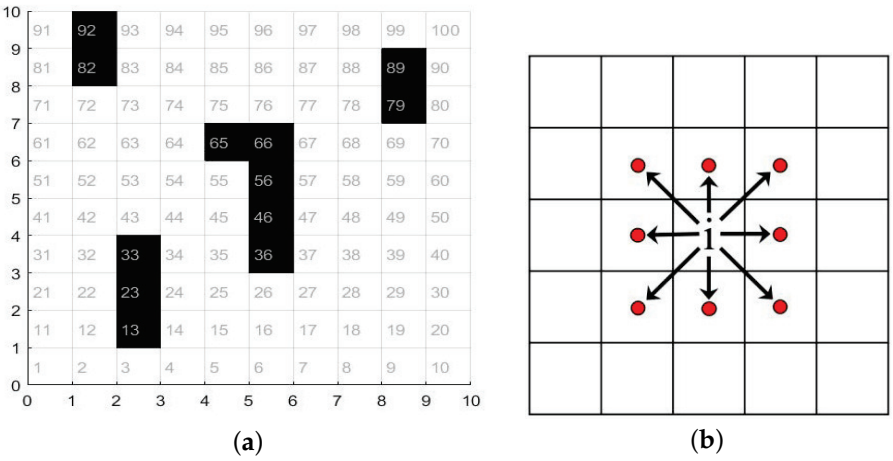

Figure 1. Raster map: (**a**) raster maps; (**b**) neighborhood matrices.

2.2. Traditional Ant Colony Optimization

The state transfer probability function in the ant algorithm primarily relies on ant colony pheromones and heuristic information [23]. In a static environment, the transfer probability $P_{ij}^{k}(t)$ for each available path node around the ith grid is computed using Equation (2). Subsequently, the roulette rule is employed to ascertain the next jth grid. The transfer probability is calculated as shown in Equation (2):

$$P_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s \in allowed_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}, j \in allowed_k \\ 0, otherwise \end{cases} \tag{2}$$

where $\tau_{ij}(t)$ represents the pheromone concentration from the ith grid to the jth grid; α is the pheromone concentration influence factor; β is the heuristic information influence factor; $allowed_k$ is the set of grids that the ants can choose; $\eta_{ij}(t)$ denotes the heuristic information from the ith grid to the jth grid, which is calculated as follows:

$$\eta_{ij}(t) = 1/d_{ij} \tag{3}$$

where d_{ij} represents the Euclidean distance [24] from the ith grid to the jth grid, as shown in the expression of Equation (4).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4}$$

After each generation of ants finishes selecting a path, the pheromone concentration is updated. The updating method is as follows:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^{m}\Delta{\tau_{ij}}^{k}, 0 < \rho < 1 \end{cases} \tag{5}$$

$$\Delta{\tau_{ij}}^{k} = \begin{cases} \frac{Q}{L_k}, Ant\ k\ traverses\ through\ the\ path\ point(i,j) \\ 0, otherwise \end{cases} \tag{6}$$

where ρ is the pheromone volatilization rate factor; $(1-\rho)$ is the ant pheromone residual factor; $\Delta{\tau_{ij}}^{k}$ is the pheromone volatilization rate coefficient of ants k released pheromone; Q is the initial value of pheromone; L_k is the total length of the ant k the total length of the path.

3. Improvement of Ant Colony Optimization

3.1. Adaptive Heuristic Functions

In traditional ant colony optimization, the heuristic information typically takes into account only the reciprocal of the distance between the current node i and the next node j [25], upon this foundation, the distance between node j and the target node g is introduced. The adaptive adjustment factor f_η of the exponential function is increased to achieve the purpose of strengthening the heuristic function's influence on the path selection in the initial stages of the algorithm and diminishing the impact of the heuristic function in the later stages, hastening the algorithm's convergence speed and enhancing the global search capability. The improvement is as follows:

$$\begin{cases} \eta_{ij}(t) = \left(\frac{1}{D}\right)^{\xi} f_\eta \\ D = \xi_1 d_{ij} + \xi_2 d_{jg} \\ d_{jg} = \sqrt{(x_j - x_g)^2 + (y_j - y_g)^2} \\ f_\eta = \left[1 - sigmoid(\frac{N_c}{N_{max}})\right] \end{cases} \tag{7}$$

where ξ is a constant greater than 1, $\xi_1 + \xi_2 = 1$; d_{jg} is the distance from node j to node g [26–28]; N_c represents the current iteration count; N_{max} denotes the maximum iteration count; the sigmoid function is a joint S-type function.

3.2. Improve Pheromone Updating Strategies

Traditional ACO uniformly updates the pheromone concentration for all ants, following Equations (5) and (6) [29]. In this paper, we introduce the strategy of elite ants updating global pheromone to enhance the pheromone of high-performing ants while attenuating the pheromone superposition of the least performing ants, reduce the interference of chaotic pheromone, and combine with the maximum–minimum ant system (MMAS), which restricts the pheromone concentration range and prevents the algorithm from falling into a premature stagnation state. The dynamically adjusted volatility coefficients are improved to prevent local optima and enhance the algorithm's efficiency in discovering the globally optimal solution. The specific improvements are as follows:

1. Pheromone update rules:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \tag{8}$$

$$\Delta\tau_{ij} = \Delta\tau_{ij}^{best} - \Delta\tau_{ij}^{worst} \tag{9}$$

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{Q_1}{L_{best}} + \frac{Q_2}{R_{best}}, The\ ant\ passes\ through\ the\ path(i,j) \\ 0, otherwise \end{cases} \tag{10}$$

$$\Delta\tau_{ij}^{worst} = \begin{cases} \frac{Q_3}{L_{worst}}, The\ ant\ passes\ through\ the\ path(i,j) \\ 0, otherwise \end{cases} \tag{11}$$

$$R_{best} = \lambda_1 Angles(L) + \lambda_2 Turns(L) \tag{12}$$

$$\lambda_1 = \frac{v}{w} \tag{13}$$

$$\lambda_2 = vt_1 \tag{14}$$

where $\Delta\tau_{ij}^{best}$ is the pheromone of the optimal ant; $\Delta\tau_{ij}^{worst}$ is the pheromone of the worst ant; Q_1, Q_2, and Q_3 are constants; L_{best}, L_{worst} represent the lengths of the optimal and worst paths, respectively; R_{best} denotes the corner conditioning factor, in which $Angles(L)$ denotes the sum of angles of the corners, and $Turns(L)$ denotes the total number of corners; λ_1 and λ_2 are the coefficient conversion expressions; v denotes the robot's constant velocity; w denotes the robot's angular velocity.

2. Pheromone interval restriction:

$$T = \begin{cases} T_{min}, T < T_{min} \\ T, T_{min} \le T \le T_{max} \\ T_{max}, T > T_{max} \end{cases} \tag{15}$$

$$\begin{cases} T_{max} = \frac{1}{2(1-\rho)L_{best}} \\ T_{min} = \frac{T_{max}}{150} \end{cases} \tag{16}$$

For any path, (i, j), the pheromone concentration T interval is restricted to $[T_{min}, T_{max}]$, as shown in Equation (15). In Equation (16), the degree of adaptivity of the algorithm is enhanced by dynamically updating T_{min} and T_{max} by extracting the optimal solution.

3. Adaptive pheromone volatilization factors:

In this paper, we adopt an adaptive adjustment strategy for ρ. At the beginning of the algorithm, the value of ρ is significant, and with an increase in the number of iterations, the value of ρ gradually decreases, reinforcing the role of pheromones in guiding the ants. The improvement formula is as follows:

$$\rho = \rho_0 L_m \rho_1 \tag{17}$$

$$\rho_1 = \left(1 + e^{\left(\frac{N_c}{N_{max}}\right)}\right)^{\mu} \tag{18}$$

where ρ_0 is taken as a constant 0.5, L_m is the path length of the mth ant, and ρ_1 is an adaptive adjustment factor for the pheromone volatilization coefficient.

3.3. Path Optimization and Deadlock Problem Handling

Due to the constraints of raster maps, the path derived from traditional ant colony algorithm planning tends to have a higher node count, the path smoothness needs to be improved, and there is an elevation in the turning angle of the path. Therefore, this paper uses a three times B spline curve to remove the redundant nodes in the path, reduce the path length, and ensure the smoothness of the final path. The comparison effect before and after optimization is shown in Figure 2a.

Due to the existence of the forbidden table, the ants may reach a deadlock state, and there is no next available grid to move. As depicted in Figure 2b, when the ant transitions from grid L to grid Z, the next available grid is V, and the ant is in a deadlock state and cannot avoid obstacles. In order to ensure the update of the global pheromone, a deadlock withdrawal mechanism is adopted, i.e., when the ant that has entered the grid is trapped in the deadlock state, the ant is allowed to return one step and update the taboo table information and pheromone update. If the ant is still deadlocked, it will continue to return until it retreats to grid L, allowing it to avoid deadlocked zones.

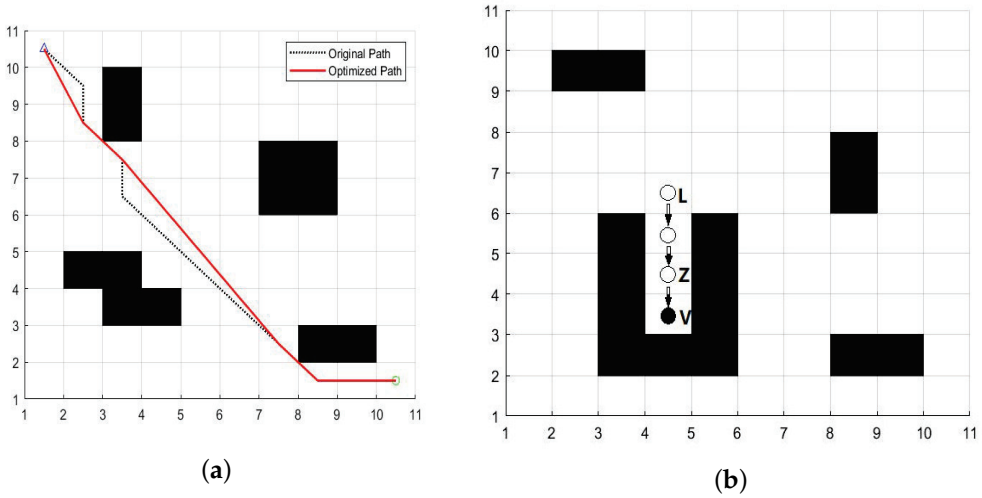

Figure 2. Path optimization: (**a**) optimizing paths using cubic B-spline curves; (**b**) deadlock state diagram.

4. Path-Planning Methods for Fusion Algorithms

4.1. Dynamic Window Approach

The DWA exhibits dynamic obstacle avoidance effectiveness by utilizing the robot detection range as a window, but the algorithm lacks global path planning guidelines; hence, the IAACO is integrated with the DWA to achieve both global path planning and local dynamic obstacle avoidance.

The DWA is a widely employed local path planning algorithm that samples multiple velocity sets in a vector space comprising velocities and accelerations integrating the constraints of velocity and acceleration; then, it models the trajectories of these velocities over a specified time interval based on the robot's motion model and evaluates the obtained trajectories according to evaluation metrics; then, it uses the corresponding velocities and accelerations as the robot's drive velocity parameters. The core of this algorithm is to transform the path-planning problem into a constrained velocity vector space optimization problem [30,31].

4.1.1. Robot Motion Model

As shown in Figure 3, the robot model used in this paper is a non-omnidirectional robot. x_{robot} and y_{robot} denote the position coordinates of the robot, and θ is the driving direction of the robot. Let $v(t)$ be the linear velocity of the robot at moment t and $\omega(t)$ be the angular velocity at moment t; the position increments of the two neighboring moments are:

$$\begin{cases} \Delta x = v_t \Delta t cos(\theta_t) \\ \Delta y = v_t \Delta t sin(\theta_t) \\ \Delta \theta = \omega_t \Delta t \end{cases} \tag{19}$$

The position at time $t+1$ is denoted as:

$$\begin{cases} x(t+1) = x(t) + v_t \Delta t cos(\theta_t) \\ y(t+1) = y(t) + v_t \Delta t sin(\theta_t) \\ \theta(t+1) = \theta(t) + \omega \Delta t \end{cases} \tag{20}$$

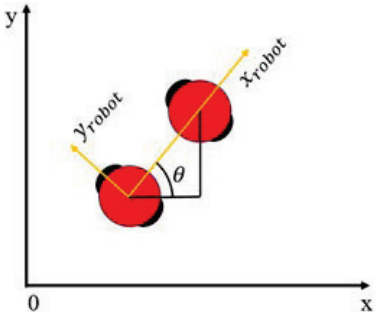

Figure 3. Motion model of the robot.

4.1.2. Robot Velocity Sampling

In the dynamic window algorithm, three constraints are introduced to delimit the range of sampled speeds. These constraints are as follows:

During the velocity sampling process, the robot's motion speed will be affected by the physical structure or environmental factors, which cannot be determined arbitrarily and is therefore subject to maximum and minimum velocity constraints, with the range of constraints as shown in Equation (21):

$$V_m = \{(v, \omega) | v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \tag{21}$$

where v_{min} and v_{max} are the robot's minimum and maximum linear velocity, respectively; ω_{min} and ω_{max} refer to the robot's minimum and maximum angular velocity, respectively.

The sampling speed of the robot needs to consider the motor performance, and due to the limited traction for acceleration and deceleration, the acceleration constraint of the robot in adjacent time needs to be set as shown in Equation (22):

$$V_d = \{(v, \omega) | v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t], \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t]\} \tag{22}$$

where v_c and ω_c represent the current linear and angular velocities; $\dot{v}_a$ and $\dot{v}_b$ denote the maximum acceleration and maximum deceleration capabilities of the robot, respectively; $\dot{\omega}_a$ and $\dot{\omega}_b$ denote the maximum angular acceleration and maximum angular deceleration, respectively.

To guarantee the safety of the robot when encountering an unforeseen obstacle, the robot must stop before hitting the obstacle; a velocity space consisting of combinations of velocities that satisfy the safety considerations of the robot's motion is proposed, as shown in Equation (23):

$$V_a = \left\{ (v,\omega) | v \le \sqrt{2dist(v,\omega)\dot{v}_b}, \omega \le \sqrt{2dist(v,\omega)\dot{\omega}_b} \right\} \tag{23}$$

where $dist(v,\omega)$ represents the minimum distance between the end of the simulated trajectory and the obstacle.

The velocity of the final dynamic window is the intersection of the above three constraints, and the velocity range of the robot can be expressed as follows:

$$V_r = V_m \cap V_d \cap V_a \tag{24}$$

4.1.3. Improvement of the Evaluation Function

Using the evaluation function, multiple trajectories are simulated based on the range of motion velocities. The robot motion model can be comprehensively evaluated, and the best trajectory can be determined based on the high score. The evaluation function of the conventional DWA is presented in Equation (25):

$$G(v,\omega) = \sigma[\delta Head(v,\omega) + \varphi Vel(v,\omega) + \psi Dist(v,\omega)] \tag{25}$$

where $Head(v,\omega)$ represents the evaluation subfunction for the directional angle deviation between the end direction of the simulated trajectory and the target point; $Vel(v,\omega)$ is the evaluation subfunction of the magnitude of the robot's running speed; $Dist(v,\omega)$ denotes the evaluation subfunction of the distance between the end of the simulated trajectory and the obstacle; δ, φ and ψ are the weights of the evaluation subfunctions; and σ is the normalization parameter of each index.

In this paper, the evaluation function is enhanced by incorporating it with critical nodes derived from global path information. This augmentation aims to increase the distance $Dist_s(v,\omega)$ between the end of the simulated trajectory and the critical point, ensuring that the final local paths are rooted in the global optimal paths. This approach facilitates the attainment of optimally smoothed paths for real-time obstacle avoidance within the global map. The improved evaluation function is shown in Equation (26):

$$G(\dot{v},\omega) = \sigma[\delta Head(v,\omega) + \varphi Vel(v,\omega) + \psi Dist(v,\omega) + \chi Dist_s(v,\omega)] \tag{26}$$

where χ is the weight coefficient.

4.2. Global Dynamic Path Planning with Fusion Algorithms

The IAACO demonstrates the capability to attain the optimal solution in the context of global path planning within a static working environment. However, it cannot avoid obstacles in time for the appearance of unknown obstacles in the environment. The DWA algorithm path planning lacks global path guidance. In an environment with more obstacles, falling into the local optimum is easy, resulting in more significant paths or path-planning failure. The two improved algorithms are fused, and the path inflection point after the path optimization of the IAACO is used as the intermediate target point of the DWA for local path planning in order to ensure the global optimum of the dynamically planned path, and while also effectively avoiding obstacles in real time. The flow of the fusion algorithm is shown in Figure 4. The specific implementation steps are as follows:

Step 1: Build an environmental grid map, initialize the initial pheromone $\tau_{ij}(0)$ of ants, the number of ants m, the initial time t, the number of iterations N_c, the maximum number of iterations N_{max}, the ant number k, and the influencing factors α,β equal parameters.

Step 2: Select the starting point S_0 of ant k and add it to the taboo table. Initialize the current coordinate point (x_0, y_0) and sample its current speed.

Step 3: Use the adaptive improved ant colony algorithm to plan a globally optimal path L_m in the grid map while determining the sub-target points (x_t, y_t) and starting point S_{dwa} of the DWA algorithm.

Step 4: Generate a simulated trajectory R based on the robot motion model for the ant coordinate points and sampling speed in step 2.

Step 5: For the path and target point, the starting point generated in Step 3, as well as the simulated trajectory generated in Step 4, select the velocity v and angular velocity ω corresponding to the optimal trajectory based on the improved evaluation function $G(v, \omega)$.

Step 6: Robot k moves along the optimal trajectory to node j and determines whether it is the target node at this time. If node j is the target node, skip to step 7. If the target node is not reached, replace the next turning point with the current target point and return to step 3.

Step 7: Determine whether the current position is the global target endpoint. If the global target endpoint is not reached, return to step 3. If the global target endpoint is reached, end the loop.

In summary, the fusion of IAACO and DWA enables robots to quickly and effectively plan paths in dynamic environments, while avoiding obstacles in real time, thereby improving the efficiency and safety of path planning.

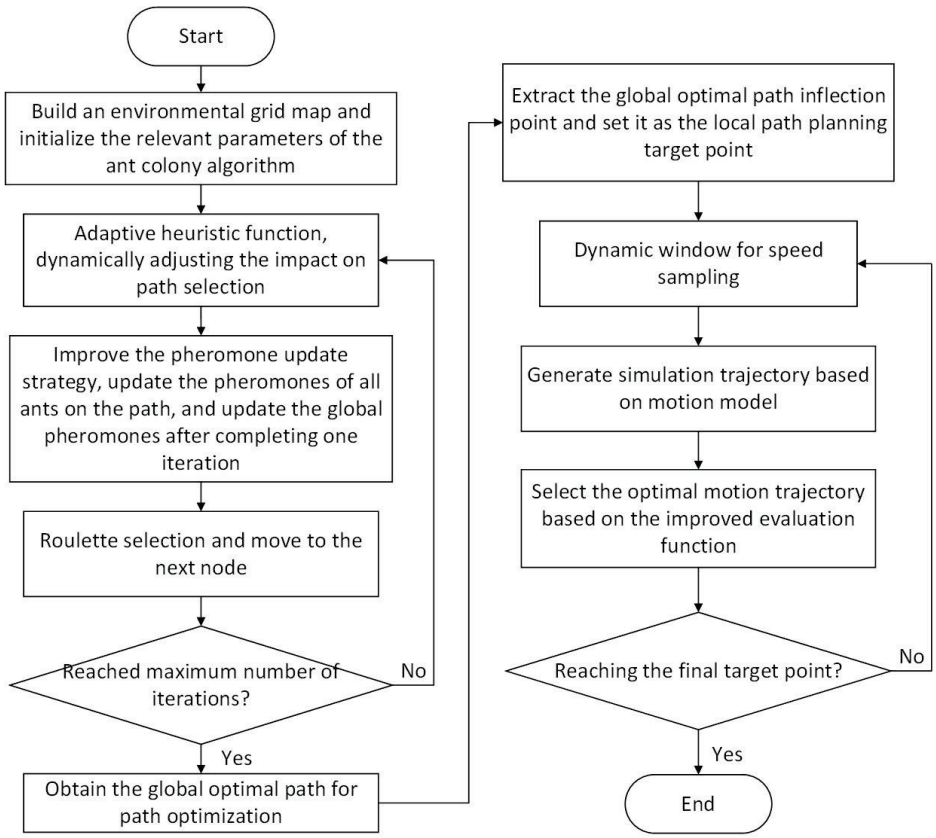

Figure 4. Flowchart of the fusion algorithm.

5. Experimental Results and Analysis

In this section, the improved algorithm is simulated, verified, and analyzed in comparison. The running environment of the algorithm is Windows 10 64-bit; processor Intel(R) Core(TM) i5-7200U; central frequency, 2.5 GHz; memory, 12 GB; simulation software, Matlab R2018a.

5.1. Experimental Analysis of Improved Ant Colony Optimization

Based on the construction basis of the environment model above, 20 × 20 and 30 × 30 raster maps with different obstacle complexity were selected for simulation comparison experiments. All the experiments are carried out independently 20 times under the same parameters, and the average value is taken for comparison. The green circle indicates the starting position, and the red triangle indicates the target position.

5.1.1. The 20 × 20 Environment

To evaluate the performance of the enhanced adaptive ant colony algorithm proposed in this study, experiments were conducted comparing it against several algorithms: ACO, the IACO algorithm presented by Luo et al. [12]; the AACO algorithm proposed by Li et al. [14]; the IAACO algorithm introduced in this paper. The experiments were carried out in a 20 × 20 grid environment map. The critical experimental parameters are detailed in Table 2, while the simulation results are summarized in Table 3. The path trajectories are illustrated in Figure 5, and the iteration count curves are depicted in Figure 6. It is worth mentioning that the experimental data from Luo et al. [12] and Li et al. [14] are utilized in their original form for comparative analysis.

Table 2. Main parameters of the experiment.

Parameter	Value	Parameter	Value
m	50	ζ_2	0.8
N_{max}	150	Q_1	1.2
α	1	Q_2	1.5
β	7	Q_3	0.8
ρ	0.3	δ	0.1
Q	100	ϕ	0.3
ζ	3	ψ	0.2
ζ_1	0.2	χ	0.2

Table 3. Comparison of simulation result data of the algorithm in 20 × 20 environment.

Algorithm	Optimal Path Length	Worst Path Length	Number of Iterations	Number of Inflection Points	Running Time
ACO	29.00	44.50	37.50	8.10	11.84
AACO	27.58	42.34	28.85	8.00	8.56
IACO	27.13	40.75	16.10	8.00	7.63
IAACO	26.07	40.40	13.75	7.55	7.79

The results of this experiment show that, after 37.50 iterations, ACO took 11.84 s to obtain the optimal path of 29.00, with 8.10 inflection points and the worst path of 44.50 in all iterations. The IACO of Li et al. [14] underwent 28.85 iterations and took 8.56 s to obtain the optimal path of 27.58, with 8.00 inflection points. Among all iterations, the worst path was 42.34. The IACO of Luo et al. [12] underwent 16.10 iterations and took 7.63 s to obtain the optimal path of 27.13, with 8.00 inflection points. Among all iterations, the worst path was 40.75. The IAACO in this article underwent 13.75 iterations and took 7.79 s to obtain the optimal path of 26.07, with 7.55 inflection points. Among all iterations, the worst path was 40.40. In a 20 × 20 environment, the optimal path length of our algorithm is 10.1%, 5.5%, and 3.9% longer than the optimal path obtained by ACO, AACO, and IACO, respectively. The corresponding number of iterations is optimized by 63.3%, 52.3%, and 14.6%, respectively. The number of inflection points in our algorithm is slightly improved compared to the other three algorithms. Regarding worst-case path and runtime, our algorithm improves by 9.2%, 4.6%, 0.9%, and 34.2%, 9%, and −2.1%, respectively, compared to the other three algorithms.

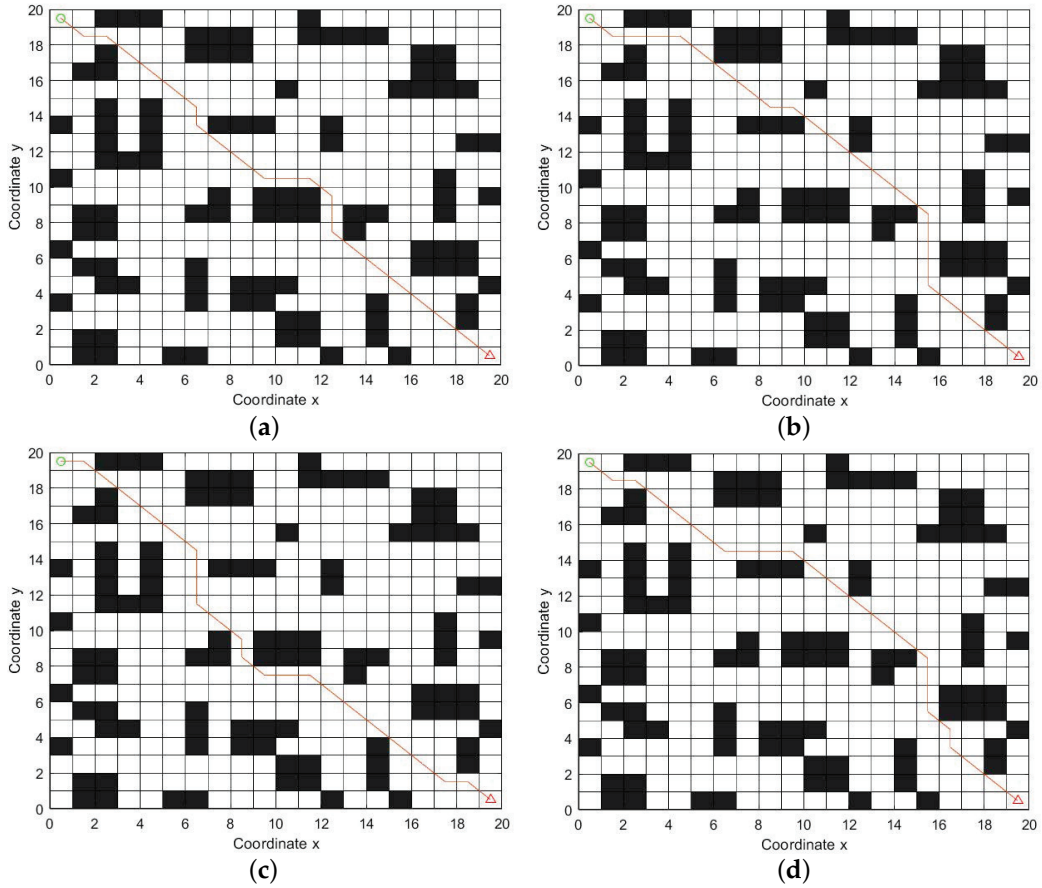

Figure 5. Path trajectory in 20 × 20 environment: (**a**) ACO; (**b**) AACO; (**c**) IACO; (**d**) IAACO.

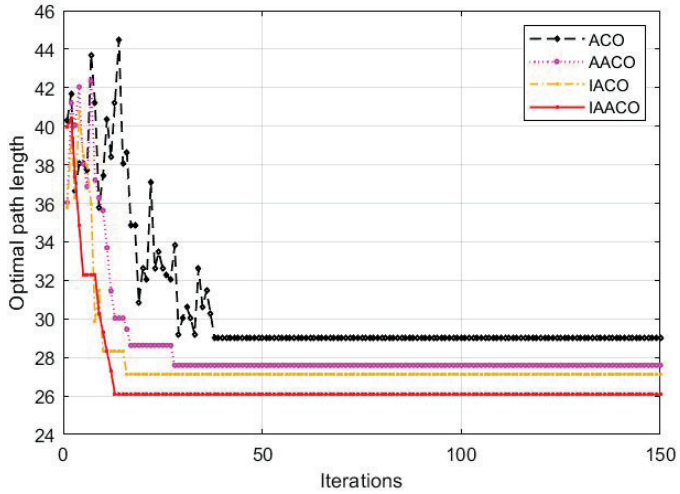

Figure 6. Trend in the number of iterations in 20 × 20 environment.

5.1.2. The 30 × 30 Environment

This article compares four algorithms through experiments on a 30 × 30 grid environment map, and the simulation results are summarized in Table 4. The path trajectory is shown in Figure 7, and the iteration count curve is shown in Figure 8.

The results of this experiment indicate that, after 62.55 iterations, ACO obtained the optimal path of 46.19 in 46.53 s, with 9.35 inflection points and 83.55 worst-case paths in all iterations. The IACO of Luo et al. [12] underwent 25.10 iterations and took 25.82 s to obtain the optimal path of 41.36 with 8.00 inflection points. In all iterations, the worst path is 55.46. The AACO of Li et al. [14] underwent 32.20 iterations and took 29.98 s to obtain the optimal

path of 44.02 with 8.00 inflection points. In all iterations, the worst path is 64.96. The IAACO in this article underwent 23.15 iterations and took 24.95 s to obtain the optimal path of 39.84 with 7.85 inflection points. In all iterations, the worst path is 52.51. In a 30 × 30 environment, the optimal path length of our algorithm is 13.7%, 9.5%, and 3.7% longer than the optimal path obtained by ACO, AACO, and IACO, respectively. The corresponding number of iterations was optimized by 63.0%, 28.1%, and 7.8%, respectively. Compared with the other three algorithms, our algorithm has slightly improved the number of inflection points. Compared with the other three algorithms, our algorithm has improved the worst-case path and runtime by 37.2%, 19.2%, and 5.3% and 46.4%, 16.8%, and 3.4%, respectively.

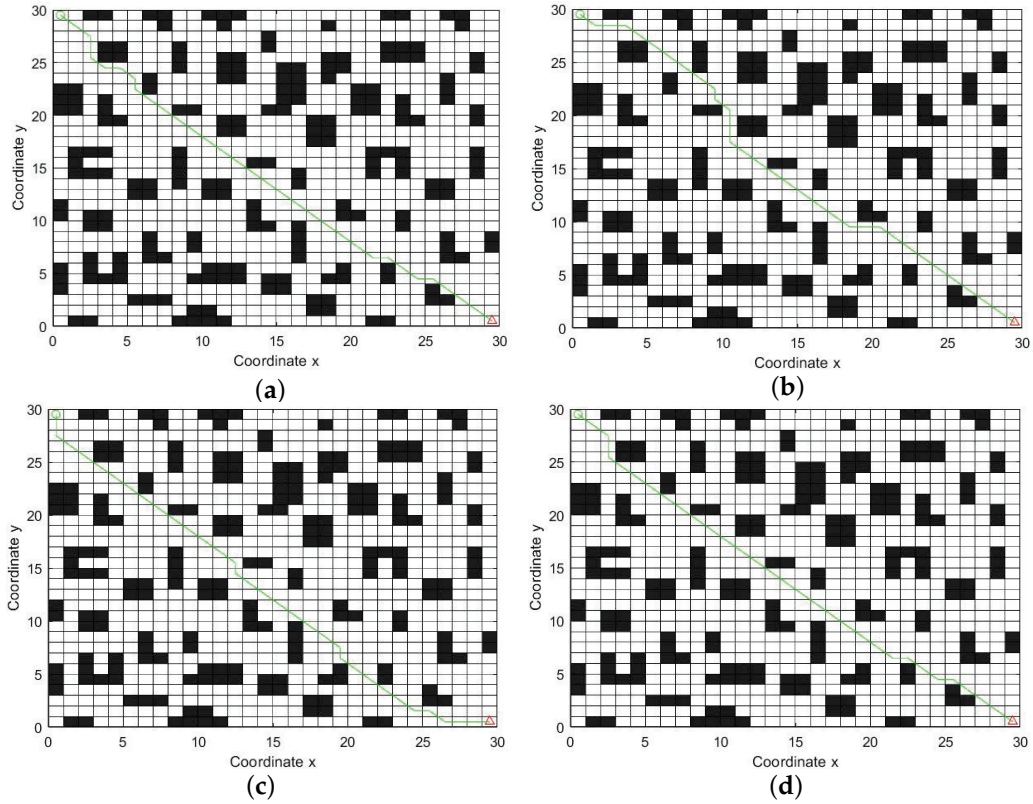

Figure 7. Path trajectory in 30 × 30 environment: (**a**) ACO; (**b**) AACO; (**c**) IACO; (**d**) IAACO.

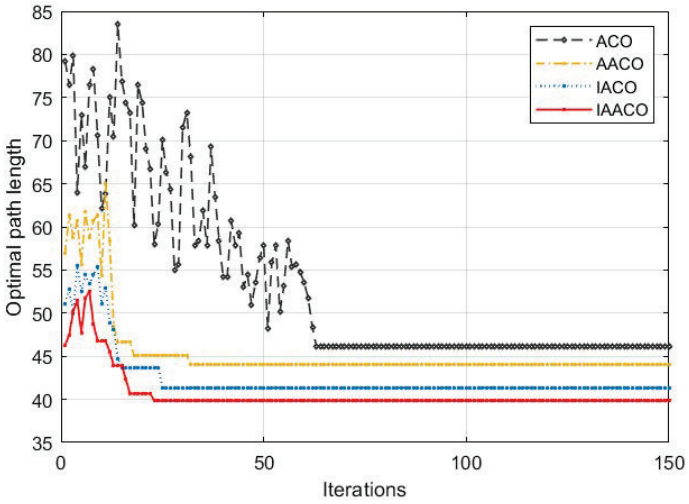

Figure 8. Trend in the number of iterations in 30 × 30 environment.

Table 4. Comparison of simulation result data of the algorithm in 30 × 30 environment.

Algorithm	Optimal Path Length	Worst Path Length	Number of Iterations	Number of Inflection Points	Running Time
ACO	46.19	83.55	62.55	9.35	46.53
AACO	44.02	64.96	32.20	8.05	29.98
IACO	41.36	55.46	25.10	8.00	25.82
IAACO	39.84	52.51	23.15	7.85	24.95

5.2. Experimental Analysis of Fusion Algorithm Obstacle Avoidance

To validate the feasibility and obstacle avoidance accuracy of the integrated algorithm in dynamic environments, path-planning experiments were conducted in a 20 × 20 grid environment using the improved ant colony algorithm and the fusion algorithm, and the parameter settings for the optimal value function of DWA (refer to Table 2). In this paper, three sets of simulation experiments were set up, namely placing one random static obstacle and one dynamic obstacle (Scenario 1), placing two random static obstacles and two dynamic obstacles (Scenario 2), and placing three random static obstacles and three dynamic obstacles (Scenario 3); here, the white circle indicates the robot model, the gray rectangle indicates the random static obstacles, and the yellow, purple, and blue rectangles indicate the dynamic obstacles. The robot moves from the start position to the end position and stops at the original place at a uniform speed. The coordinate information of each obstacle is shown in Table 5, the simulation experiment results of the fusion algorithm are shown in Figure 9, and the feedback results of the robot control parameters are shown in Figure 10.

Table 5. Random static and dynamic obstacle coordinate parameters.

Obstacle	Starting Coordinates	End Point Coordinates
D1	[3.5, 10.5]	[5.5, 19.5]
D2	[6.5, 3.5]	[16.5, 18.5]
D3	[13.5, 2.5]	[15.5, 13.5]
J1	[7.5, 15.5]	-
J2	[11.5, 12.5]	-
J3	[16.5, 6.5]	-

As shown in Figure 9, the blue dashed line and the blue solid line represent the IAACO route and the fusion algorithm route, respectively, and the black dashed line represents the dynamic obstacle running route. The robot runs according to the critical points of the IAACO planning route before encountering random static and dynamic obstacles. When random static and dynamic obstacles are detected, it predicts by calculating the relative motion state, whether a side collision, a frontal collision, etc., to carry out local obstacle avoidance processing according to the fusion algorithm and, after avoiding the obstacles, continuously re-plan the route to reach the target point based on the optimal path planned by the IAACO. Figure 9a,d,g shows that the robot successfully avoids dynamic obstacles D1, D2, and D3, respectively; Figure 9b,e,h shows that the robot successfully avoids random static obstacles J1, J2, and J3, respectively. Figure 9c,f,i shows that the robot reaches the goal point after completing the avoidance process in the three scenarios.

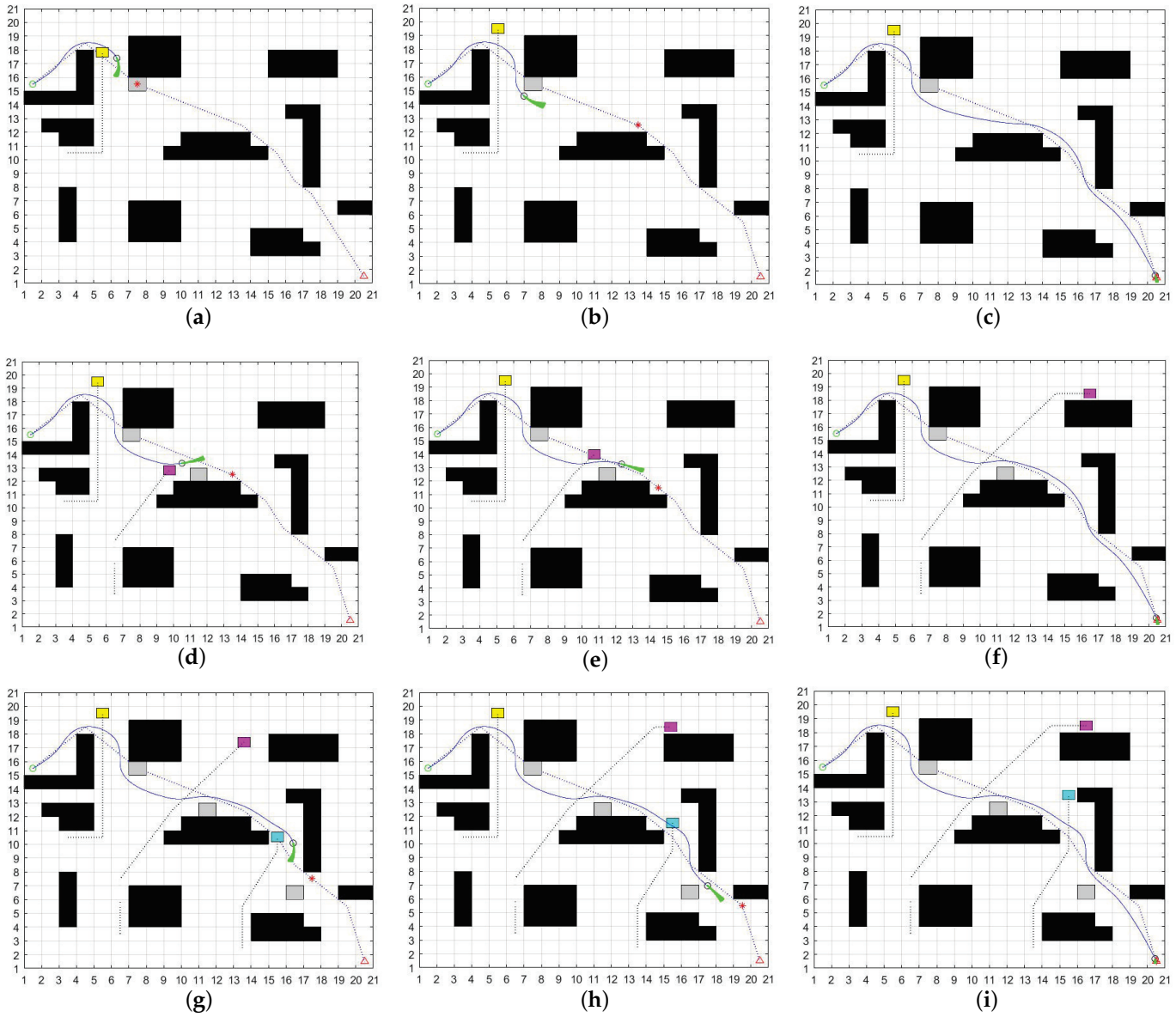

Figure 9. Dynamic obstacle avoidance process: (**a**–**c**) Scenario 1 realization process; (**d**–**f**) Scenario 2 realization process; (**g**–**i**) Scenario 3 realization process.

5.3. Experimental Analysis of Real Environment Planning

In this paper, the robot platform in Figure 11 is used to verify the feasibility and adaptability of the algorithm in the natural environment. The robot industrial control machine is the host, the personal PC is the secondary; the primary and the secondary are installed the same operating system, through the remote control in order to complete the control operation of the secondary to the host. The use of the host's ROS system occurs under the Rviz and Gazebo visualization tools, and according to the Gmapping algorithm for slam mapping and the point cloud data information obtained by the sensor are used to complete the construction of the experimental environment. The ROS terminal of the host computer is opened through the secondary; the LIDAR is started to release the coordinate transformation node. The Gmapping map building node is started, and the joystick is used to remotely control the robot trolley to walk around in the experimental scene shown in Figure 12c. The scene map is shown in Figure 12a. This should be displayed under the visualization interface of Rviz; then, the starting target point should be determined; the improved algorithm should be started based on this paper's navigation algorithm for path planning. The green solid line in this figure indicates the planned path and Figure 12b

indicates that the robot model carries out autonomous navigation according to this route. Figure 12c–f indicates that the robot trolley completes the running process from the starting point to the target point, as well as avoiding obstacles 1–4, respectively, based on the route of the path planning.

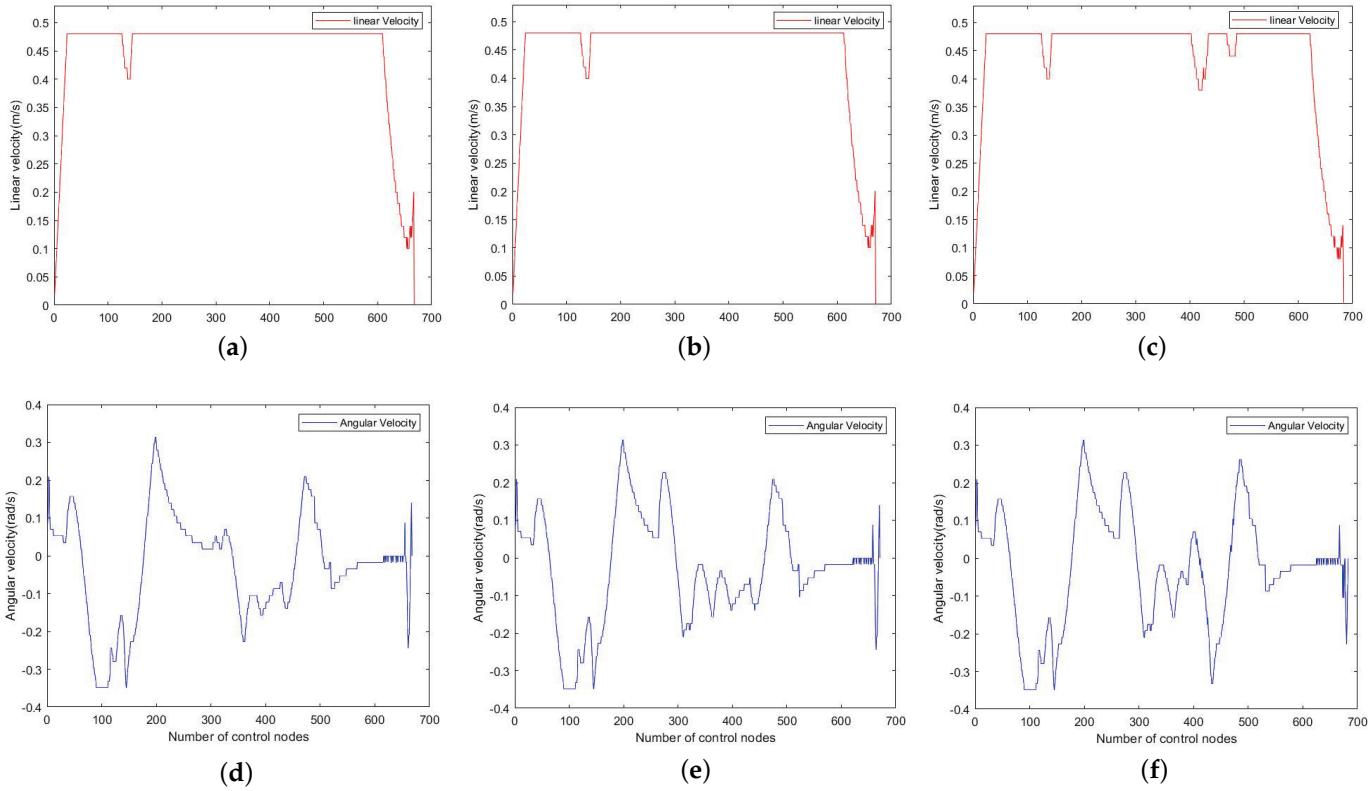

Figure 10. Robot output results for 3 scenarios: (**a**–**c**) Linear velocity output results; (**d**–**f**) Angular velocity output results.



Figure 11. Actual experiment platform.

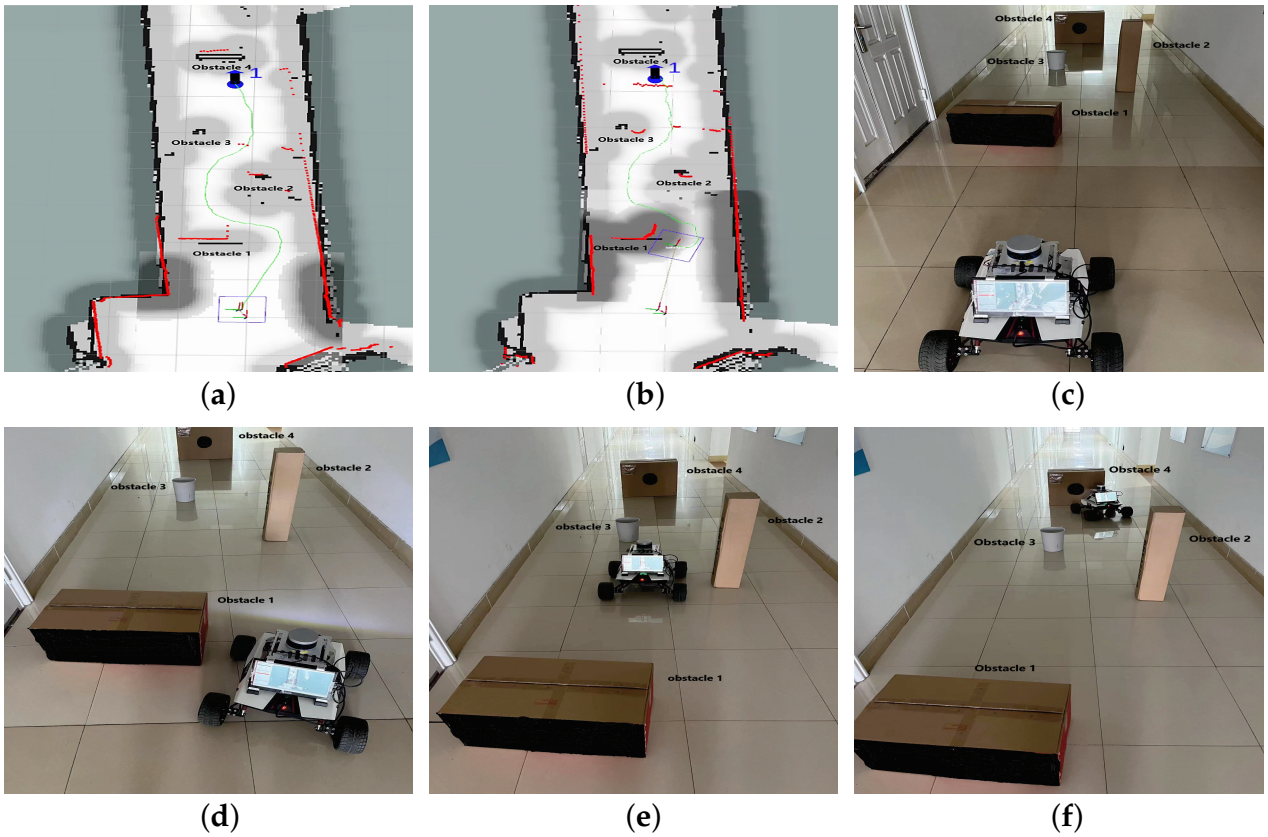

Figure 12. Obstacle avoidance live test: (**a**) global path planning; (**b**) running the avoidance route; (**c**) starting point; (**d**) obstacle avoidance; (**e**) obstacle avoidance 2 and 3; (**f**) end of obstacle.

6. Conclusions

This article addresses the many limitations of traditional ant colony algorithm (ACO) in global path planning, such as path redundancy, node redundancy, slow convergence speed, and lack of real-time random obstacle avoidance ability. A fusion algorithm combines an enhanced ant colony optimization algorithm (IAACO) and dynamic window approach (DWA). The main improvements and contributions of this article include:

1. Correction of heuristic information distance function: By adding an adaptive adjustment factor to the exponential function, the convergence speed of the algorithm is accelerated, and the efficiency of path planning is improved.
2. Improve the pheromone update strategy: Add elite ants and max–min ants systems and introduce an adaptive pheromone volatilization factor for dynamic adjustment to enhance the adaptability and convergence of the algorithm.
3. Path optimization based on cubic B-splines: reducing node redundancy and enhancing path smoothness through path optimization, thereby improving the quality of path planning.
4. Introducing an exit mechanism to avoid ants becoming stuck and ensure that the algorithm can quickly converge to the global optimal solution.
5. Enhance the evaluation function of the dynamic window method: improve the robots' real-time random obstacle avoidance ability in local planning and increase the applicability and flexibility of the algorithm.

Through simulation experiments under different environmental information, the improved algorithm in this paper can plan shorter paths, reduce iteration times, and achieve global optimal path planning in static and dynamic obstacle environments. In addition, this article also conducted experimental verification in natural environments, further proving the effectiveness and practical value of the fusion algorithm. However, we are also aware of some limitations in this study. The experimental data are only based

on specific simulation environments and may be affected by more complex real-world situations in practical applications. The robustness and applicability of the algorithm still need to be further improved. Future research can further explore the application of algorithms in more complex environments and tasks, optimize the parameter settings of algorithms, and further improve the robustness and scalability of algorithms. It is also possible to further explore the integration of more robot path-planning algorithms to improve the efficiency and accuracy of path planning.

Author Contributions: Conceptualization, D.S. and S.Z.; methodology, X.W. and P.Z.; software, S.Z.; data curation, S.Z.; writing—original draft preparation, S.Z.; writing—review and editing, D.S.; supervision, D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Key R&D Plan of Shandong Province (Major Science and Technology Innovation Project) No. 2023CXGC010701; in part by the Project of 20 Policies of Facilitate Scientific Research in Jinan Colleges No. 2019GXRC063; in part by the Project of Shandong Provincial Major Scientific and Technological Innovation, grant No.2019JZZY010444, No. 2019TSLH0315; and in part by the Natural Science Foundation of Shandong Province, grant No. ZR2020MF138.

Data Availability Statement: The data presented in this study are available in this article.

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

References

1. Chen, Y.; Gao, A. Ant Colony Planning Strategy for Multiple Swarm Games of Robot Navigation Paths. *Mech. Des. Manuf.* **2021**, *1*, 272–276+281.
2. Zhang, W.; Gong, X.; Han, G. An improved ant colony algorithm for path planning in one scenic area with many spots. *IEEE Access* **2017**, *5*, 13260–13269. [CrossRef]
3. Shi, Y.; Zhang, H.; Li, Z.; Hao, K.; Liu, Y.; Zhao, L. Path planning for mobile robots in complex environments based on improved ant colony algorithm. *Math. Biosci. Eng.* **2023**, *20*, 15568–15602. [CrossRef] [PubMed]
4. Singh, A.; Shakeel, M.; Kalaichelvi, V.; Karthikeyan, R. A Vision-Based Bio-Inspired Reinforcement Learning Algorithms for Manipulator Obstacle Avoidance. *Electronics* **2022**, *11*, 3636. [CrossRef]
5. Li, K.; Hu, Q.; Liu, J. Path Planning of Mobile Robot Based on Improved Multiobjective Genetic Algorithm. *Wirel. Commun. Mob. Comput.* **2021**, *9*, 2021. [CrossRef]
6. Kang, J.; Lim, D.; Choi, Y.; Jang, W.; Jung, J. Improved RRT-Connect Algorithm Based on Triangular Inequality for Robot Path Planning. *Sensors* **2021**, *2*, 333. [CrossRef] [PubMed]
7. Eshtehardian, S.; Khodaygan, S. A continuous RRT*-based path-planning method for non-holonomic mobile robots using B-spline curves. *J. Ambient Intell. Human. Comput.* **2023**, *14*, 8693–8702. [CrossRef]
8. Zhang, H.; Zhu, Y.; Liu, X.; Xu, X. Analysis of Obstacle Avoidance Strategy for Dual-Arm Robot Based on Speed Field with Improved Artificial Potential Field Algorithm. *Electronics* **2021**, *10*, 1850. [CrossRef]
9. Zhang, B.; Yang, Y.; Zhu, H.; Liu, A.; Ni, H. Dynamic path planning algorithm for mobile robots based on deep reinforcement learning. *Comput. Meas. Control* **2023**, *31*, 153–159+166.
10. Lei, T.; Ming, L. A robot exploration strategy based on Q-learning network. In Proceedings of the IEEE International Conference on Real-Time Computing and Robotics (RCAR), Angkor Wat, Cambodia, 6–10 June 2016; pp. 57–62.
11. Liu, S.; Huang, Y. Application of Multi Strategy Ant Colony Algorithm in Robot Path Planning. *Comput. Eng. Appl.* **2022**, *58*, 278–286.
12. Luo, Q.; Wang, H.; Zheng, Y. Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Comput. Appl.* **2020**, *32*, 1555–1566. [CrossRef]
13. Tian, X.; Liu, L.; Liu, S.; Du, Z.; Pang, M. Path planning of mobile robot based on improved ant colony algorithm for logistics. *Math. Biosci. Eng.* **2021**, *18*, 3034–3045.
14. Li, Y.; Ming, Y.; Zhang, Z.; Yan, W.; Wang, K. An Adaptive Ant Colony Algorithm for Autonomous Vehicles Global Path Planning. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021.
15. Chen, Y.; Wu, J.; He, C.; Zhang, S. Intelligent Warehouse Robot Path Planning Based on Improved Ant Colony Algorithm. *IEEE Access* **2023**, *11*, 12360–12367. [CrossRef]
16. Si, M.; Zhou, X.; Zhang, Y. Improvement of Dynamic Window Approach in Dynamic Obstacle Environment. *J. Phys. Conf. Ser.* **2023**, *1*, 012059. [CrossRef]
17. Wang, Z.; Liang, Y.; Gong, C.; Zhou, Y.; Zeng, C.; Zhu, S. Improved Dynamic Window Approach for Unmanned Surface Vehicles' Local Path Planning Considering the Impact of Environmental Factors. *Sensors* **2022**, *22*, 5181. [CrossRef]
18. Jin, Q.; Tang, C.; Cai, W. Research on Dynamic Path Planning Based on the Fusion Algorithm of Improved Ant Colony Optimization and Rolling Window Method. *IEEE Access* **2022**, *10*, 28322–28332. [CrossRef]

19. Wang, Q.; Li, J.; Yang, L.; Yang, Z.; Li, P.; Xia, G. Distributed Multi-Mobile Robot Path Planning and Obstacle Avoidance Based on ACO–DWA in Unknown Complex Terrain. *Electronics* **2022**, *11*, 2144. [CrossRef]
20. Xiong, J.; Huang, H. Path planning for mobile robots by integrating improved ant colony and DWA. *Mach. Des. Manuf.* **2023**, *6*, 289–295.
21. Zhao, Q.; Huang, Y. Robot path planning using A * ant colony and dynamic window method. *J. Electron. Meas. Instrum.* **2023**, *37*, 28–38.
22. Li, X.; Li, Q.; Zhang, J. Research on global path planning of unmanned vehicles based on improved ant colony algorithm in the complex road environment. *Meas. Control* **2022**, *55*, 945–959. [CrossRef]
23. Zheng, Y.; Luo, Q.; Wang, H.; Wang, C.; Chen, X. Path Planning of Mobile Robot Based on Adaptive Ant Colony Algorithm. *J. Intell. Fuzzy Syst.* **2020**, *39*, 5329–5338. [CrossRef]
24. Mazen Hittawe, M.; Sidibé, D.; Beya, O.; Mériaudeau, F. Machine vision for timber grading singularities detection and applications. *J. Electron. Imaging* **2017**, *26*, 063015.
25. Yang, L.; Fu, L.; Li, P.; Mao, J.; Guo, N. An Effective Dynamic Path Planning Approach for Mobile Robots Based on Ant Colony Fusion Dynamic Windows. *Machines* **2022**, *10*, 50. [CrossRef]
26. Tahri, O.; Usman, M.; Demonceaux, C.; Fofi, D.; Hittawe, M. Fast earth mover's distance computation for catadioptric image sequences. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2485–2489.
27. Zhang, Y.; Hittawe, M.; Katterbauer, K.; Marsala, A.F.; Knio, O.M.; Hoteit, I. Joint seismic and electromagnetic inversion for reservoir mapping using a deep learning aided feature-oriented approach. In *SEG Technical Program Expanded Abstracts*; Society of Exploration Geophysicists: Tulsa, OK, USA, 2020; pp. 2186–2190.
28. Hittawe, M.; Afzal, S.; Jamil, T.; Snoussi, H.; Hoteit, I.; Knio, O. Abnormal events detection using deep neural networks: Application to extreme sea surface temperature detection in the Red Sea. *J. Electron. Imaging* **2019**, *28*, 021012. [CrossRef]
29. Wu, L.; Huang, X.; Cui, J.; Liu, C.; Xiao, W. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. *Expert Syst. Appl.* **2023**, *215*, 119410. [CrossRef]
30. Lai, X.; Wu, D.; Wu, D.; Li, J.; Yu, H. Enhanced DWA algorithm for local path planning of mobile robot. *Ind. Robot.* **2023**, *50*, 186–194. [CrossRef]
31. Zeng, D.; Chen, H.; Yu, Y.; Hu, Y.; Deng, Z.; Zhang, P.; Xie, D. Microrobot Path Planning Based on the Multi-Module DWA Method in Crossing Dense Obstacle Scenario. *Micromachines* **2023**, *14*, 1181. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.