Mathematical Biosciences and Engineering

MBE, 20(11): 19152–19173.
DOI: 10.3934/mbe.2023846
Received: 28 June 2023
Revised: 13 August 2023
Accepted: 21 September 2023
Published: 13 October 2023

http://www.aimspress.com/journal/MBE

Research article

Research and experiment on global path planning for indoor AGV via improved ACO and fuzzy DWA

Zhen Zhou*, Chenchen Geng, Buhu Qi, Aiwen Meng and Jinzhuang Xiao

Key Laboratory of Digital Medical Engineering of Hebei Province, College of Electronic and Information Engineering, Hebei University, Baoding 071000, China

* **Correspondence:** Email: 1006300202@qq.com.

Abstract: In order to obtain an optimal trajectory for indoor AGV, this paper combined an improved ACO and fuzzy DWA (IACO-DWA) algorithm, which can provide an optimal path with collision-free under higher optimization efficiency. The highlights of this paper are detailed as follows: Firstly, an improved adaptive pseudo-random transition strategy is adopted in the state transition probability with an angle factor. A reward and punishment mechanism is introduced in the pheromone updating strategy, then a path optimization strategy called IACO is proposed for the more optimized path. Secondly, IDWA adopted three fuzzy controllers of direction, security and adjustment coefficients through evaluating directional and safety principles, then improving the angular velocity by processing the linear velocity with linear normalization. By adapting to the changes of the environment, the IDWA parameters can be dynamically adjusted to ensure the optimal running speed and reasonable path of AGV. Thirdly, aiming to deal with the path-planning problem in complex environments, we combined IACO with IDWA, the hybrid algorithm involves dividing the globally optimal path obtained from IACO planning into multiple virtual sub-target points. IDWA completes the path planning by switching between these local target points, thereby improving the efficiency of the path planning. Finally, simulations is verified by Matlab and experiment results on the QBot2e platform are given to verify IACO-DWA algorithm's effectiveness and high performance.

Keywords: ant colony algorithm; fuzzy control; global path planning; AGV; dynamic window approach

1. Introduction

With the paradigm shift of Industry 4.0 and the rapid development of automation technology, automatic guided vehicles (AGVs) is a mobile robot widely used in manufacturing facilities, warehouses to transport goods from starting point to target point without any manual intervention. The COVID-19 outbreak created economic turmoil for almost every industry at the global level, which has resulted in a vast decline in profits and revenue for the manufacturing industry. AGVs can deliver goods without direct contact of human beings, greatly reducing the risk of epidemic transmission while increasing work efficiency. It plays a crucial role in ensuring that businesses meet certain production levels during such turmoil [1,2]. A global automated guided vehicle market size was valued at USD 4.28 billion in 2022 and is expected to grow at a compound annual growth rate (CAGR) of 9.7% from 2023 to 2030 [3]. In brief, AGVs has a broad prospect and trends.

Mobile robot is mainly divided into three parts: information perception, path planning and motion control. Path planning is a bridge between information perception and motion control, and it is an important part of AGV to complete transportation tasks. The development of path planning technology provides great convenience for mankind. For example, the application of AGV can greatly reduce labor cost and improve efficiency. In disaster relief, the application of AGV can go deep into the disaster center to accurately quickly obtain information reduce casualties and property safety. All of these applications employ path planning techniques. Therefore, in view of the application prospect of path planning technology, more and more scholars are studying it.

In recent years, in order to achieve the higher requirements of AGV path planning, the researchers have proposed many algorithms to solve the path planning problem of robots. However, most scholars proposed that the algorithm is a single algorithm, and its limitations cannot consider the indicators such as path security and smoothness. This paper aims to solve the problems of insufficient smooth path, insufficient security and too long path in indoor AGV path planning and proposes an efficient path planning algorithm (IACO-DWA) by combining global and local path planning algorithms. It mainly has the following contributions:

1) In Improving Ant Colony Optimization (IACO), we improve the state transition probability by adding new angle factors and adopting an adaptive pseudo-random transition strategy, then introducing a punishment mechanism in the pheromone to improve the efficiency of the algorithm. Furthermore, we propose a path optimization strategy based on the optimal path to further minimize the path length.

2) In Improving Dynamic Window Approach (IDWA), we address the issue of changing environmental complexity during AGV operation by incorporating fuzzy control. Specifically, we add fuzzy control to the original DWA and adjust the weight parameters of the DWA based on the obstacle environment complexity. Additionally, we utilize linear velocity and diagonal velocity adjustments to further optimize the algorithm performance.

3) We divide the global path obtained through IACO planning into multiple key nodes, which serve as local target points for IDWA to optimize AGV path planning. We verify the effectiveness and practicality of the proposed IACO-DWA algorithm by implementing simulations in Matlab and experiments on the QBot2e platform.

This paper is structured as follows: Section 2 introduces the related work. Section 3 provides an introduction to IACO, IDWA, and IACO-DWA algorithms. In Section 4, we present the simulation results of the above three algorithms. Section 5 validates the effectiveness and utility of the proposed IACO-DWA algorithm. Finally, in Section 6, we conclude the paper and discuss future work.

2. Related works

Path planning algorithm is generally divided into global path planning algorithm and local path planning algorithm according to different environmental information [4]. Global path planning refers to planning a path from the starting point to the target point with known environment information. Common global path planning algorithms, including the D* algorithm [5] and RRT algorithm [6]. In recent years, more and more scholars are studying the intelligent optimization algorithm because of its strong robustness advantage. These intelligent optimization algorithms are widely used in image processing, diagnostic disease, and path planning [7,8]. The latest proposed optimization algorithm including Harris Hawk Optimization (HHO) algorithm [9], Monarch Butterfly Optimization (MBO) [10], Colony Predation algorithm (CPA) [11]and Ant Colony Optimization (ACO) [12]. However, ACO is widely used in AGV path planning due to its advantages in simplicity, robustness, and adaptability. Traditional ACO processing path planning also has some drawbacks, such as insufficient smooth path, insufficient security and too long path. Many scholars have proposed improvements to address these problems. Zhou et al. [13] introduced an ACO-DA-based optimization model for airport AGV path planning, this approach used the ACO algorithm for optimizing baggage collection and sorting, followed by the Dijkstra algorithm to plan the AGV path, which improves the efficiency of airport AGV path planning. Luo et al. [14] applied the unequal allocation of initial pheromones and dynamic punishment method to solve the problem of deadlock. Yang et al. [15] proposed distance factor heuristic and smoothing factor functions which shorten the path length and the number of turns. The existing strategies for improving ACO usually do not consider the AGV size and dynamics model, which does not meet the AGV motion constraints, and it is difficult to meet the indexes of path smoothness and safety.

Local path planning typically involves two methods: artificial potential field (APF) [16] and dynamic window approach (DWA) [17]. APF drives AGV motion through a virtual potential field. However, in complex environments, when the sum of the potential field's repulsion force and attraction vector is zero, the AGV may struggle to reach its destination and may become trapped in a local optimum. Additionally, such algorithms often do not take into account the AGV kinematic model [18]. DWA is widely regarded as a real-time obstacle avoidance local path planning method, which is generated by feasible speed combination trajectory and score, which can directly get the optimal speed and consider the physical limitations, environmental limit, and current state [19]. The common problem of DWA algorithm is that it is prone to local optima and inaccessible goals. To solve these problems, Wang et al. [20] proposed a DDPG-DWA algorithm, which converted the collision avoidance problem into the optimal reward problem to improve the path planning efficiency of DWA. Bai et al. [21] proposed a G-DWA algorithm, which is improved by the phototropism of the plant, so the Unmanned Surface Vehicles (USV) can reach the target point smoothly. Ji et al. [22] proposed a fusion of A* algorithm and DWA algorithm. Although it can meet the requirements of smoothness and security, the path is not globally optimal. The existing DWA algorithm mainly has two problems. On the one hand, DWA algorithm lacks the guidance of global path information, and is easy to fall into the local optimum. On the other hand, the evaluation function mechanism is unreasonable and the weight coefficient is constant, leading to a long planning path.

As mentioned, the shortage of existing ACO does not consider the kinetic model of AGV, and the path is difficult to meet the smoothness and safety requirements. The shortage of the existing DWA algorithm is the constant weight coefficient of the evaluation function and the lack of global information. Most of the existing path planning algorithms only consider a single path planning algorithm. With the complexity of path planning tasks, single path planning algorithms often struggle to meet the high-quality path planning requirements. Thus, aiming at shortcomings of the existing algorithms, this paper proposes an efficient path planning algorithm (IACO-DWA) for indoor AGV by combining global and local path planning algorithms. The effectiveness and utility of the IACO-DWA algorithm are verified by simulations and experiments.

3. Proposed IACO-DWA algorithm

3.1. Improvement ACO

3.1.1. Improvement state transition probability

ACO algorithm originates from the ant's natural foraging scene, where ants find the best path from the nest to the food source by releasing pheromones, if the ant finds a shorter path, it releases more pheromone to guide other ants along this path [23]. Figure 1 is a schematic diagram of ACO algorithm.

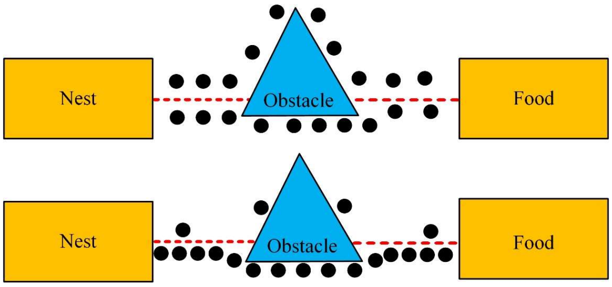


Figure 1. Schematic diagram of the principle of ant colony optimization.

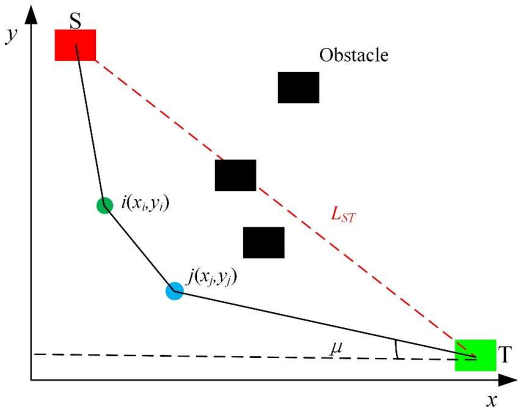


Figure 2. A schematic diagram of adding the new heuristic function μ_{ij} .

To obtain shorter paths, a new heuristic function μ_{ij} is applied to ACO state transition probability. Figure 2 illustrates the ideal path L_{ST} . The larger the value of angle μ_{ij} , the closer the path length comes to L_{ST} , and thus the shorter the overall path becomes. Eq (1) presents the angle calculation formula, while Eq (2) presents the new transition probability.

$$\mu_{ij} = sin\,\mu = \frac{\left|x_T - x_j\right|}{L_{jT}} \tag{1}$$

$$P_{ij}^k\left(\mathrm{t}\right) = \begin{cases} \dfrac{\left[\tau_{ij}\left(\mathrm{t}\right)\right]^{\alpha}\left[\eta_{ij}\left(\mathrm{t}\right)\right]^{\beta}\left[\mu_{ij}\left(\mathrm{t}\right)\right]^{\lambda}}{\sum_{s \in allowed_k}\left[\tau_{ij}\left(\mathrm{t}\right)\right]^{\alpha}\left[\eta_{ij}\left(\mathrm{t}\right)\right]^{\beta}\left[\mu_{ij}\left(\mathrm{t}\right)\right]^{\lambda}}, j \in allowed_k \\ 0, \text{otherwise} \end{cases} \tag{2}$$

where η_{ij} represents the Euclidean distance between nodes i and j . τ_{ij} denotes the pheromone concentration from node i to j . μ_{ij} refers to the angle between the connection line of the current node i and the target point T and the X-axis. $allowed_k$ represents the set of potential next nodes for the ant to move. α , β , λ are the influence factor of the pheromone concentration, the expected heuristic, and the angle respectively.

This paper adopts an improved adaptive pseudorandom transfer strategy in the transition probability of ACO. The improved state probability transfer formula is shown in Eq (3):

$$J = \begin{cases} \text{argmax}\left\{\left[\tau_{ij}\left(t\right)\right]^{\alpha}\left[\eta_{ij}\left(t\right)\right]^{\beta}\left[\mu_{ij}\left(t\right)\right]^{\lambda}\right\}, \text{if } q < q_0 \\ P_{ij}^k, \text{otherwise} \end{cases} \tag{3}$$

when $q < q_0$, state transition probability J is the maximum of the product of the heuristic function and pheromone concentration, in this case, the ant selects the node with the largest pheromone and heuristic combination.

This paper adopts the probability of the adaptive state transition parameter q_0 to select the next moving node. In the early stages of the algorithm, when q_0 is close to 1, the role of heuristic information is strengthened to improve the convergence speed of the algorithm. As the algorithm progresses, q_0 gradually decreases, and the effect of heuristic information is weakened, thus enhancing the global search ability of the algorithm. The adjustment rule is shown in Eq (4):

$$q_0 = \exp\left(-\frac{1}{2}\left(\frac{N_C}{N_{C\max}}\right)\right) \tag{4}$$

where N_C is the current number of iterations and $N_{C\max}$ is the maximum number of iterations.

3.1.2. Improvement pheromone update rule

Once all ants have completed one iteration, the pheromone update rule rewards the optimal path to enhance its guidance effect, while also weakening the pheromone concentration along the worst path and avoiding the blind updating of the global pheromone. The local pheromone rule is updated using Eq (5), while global pheromone update rules are shown in Eqs (6) and (7):

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{Q}{L_{best}},(i,j)\in L_{best}\\ 0,\text{otherwise}\\ -\dfrac{2Q}{L_{best}+L_{worst}},(i,j)\in L_{worst}\end{cases} \tag{5}$$

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\left(\frac{Q}{L_{best}}\right),ij\in L_{best} \tag{6}$$

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\left(-\frac{2Q}{L_{best}+L_{worst}}\right),ij\in L_{worst} \tag{7}$$

where $\Delta\tau_{ij}(t)$ represents the pheromone increment sum of all ants, $\tau_{ij}(t+1)$ represents the updated information quantity, $\rho\in(0,1)$ is the global pheromone volatilization coefficient, Q represents the initial value of pheromone intensity, L_{best} denotes the length of the current optimal path, L_{worst} denotes the length of the worst path in the current full path.

This paper adopts an adaptive pheromone volatilization coefficient. During the early stages of the algorithm, the volatile coefficient increases path diversity. In the later stages, the volatile coefficient becomes larger, accelerating the convergence of the algorithm. The adaptive pheromone volatilization coefficient is formulated as shown in Eq (8).

$$\rho=\rho_{\min}+(\rho_{\max}-\rho_{\min})*\left(\frac{N_C}{N_{C\max}}\right)*\left(\frac{L_{sg}}{L_{gb}}\right) \tag{8}$$

where $\rho_{\min}$ and $\rho_{\max}$ are the minimum and maximum values of the volatilization coefficient, respectively, N_C represents the current number of iterations, $N_{C\max}$ represents the maximum number of iterations, L_{sg} represents the Euclidean distance from the starting point to the target, and L_{gb} represents the global optimal path.

3.1.3. Path optimization strategy

Traditional path planning involves connecting the center points of grids which results in the path often contains redundant nodes, it is not align with the actual AGV path planning. To address this problem, the paper proposes a key node path optimization strategy that can smooth and optimize the path, eliminating the need to limit path selection to the central position of each grid, The path smoothing optimization process is illustrated in Figure 3. Using this path as an example, the optimization steps are as follows:

Step 1: Enter ACO to obtain the optimal set of path points as $U = \{S, A, B, C, D, E, F, G, H, I, J, K, T\}$;

Step 2: Starting from the starting point S, sequentially connect path points in set U. If the line between two connected points intersects an obstacle, designate both the current point and the previous point as key nodes. Designate the current point as the new starting node and continue connecting subsequent points until reaching the target endpoint T. Then add the starting point, the end point, and all key nodes to set $Q = \{S, B, C, G, H, T\}$.

Step 3: Connect the key nodes in the set Q in turn, and the algorithm ends.

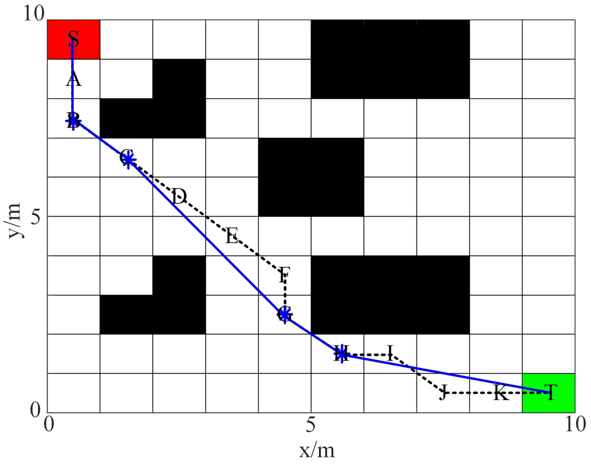


Figure 3. A Schematic of the path-smoothing optimization.

3.2. Improvement DWA

Dynamic window method is the latest research method in the field of motion planning algorithms, which proposes a real-time collision avoidance strategy for AGV. DWA algorithm relies on the dynamics of AGV to design the optimal avoidance strategy while taking into account the constraints on AGV motion imposed by various parameters such as velocity and acceleration. Based on the range of motion speed and the robot motion model, the algorithm selects the optimal trajectory through an evaluation function, as shown in Eq (9).

$$G(v, \omega) = \delta\left(\alpha \cdot P(v, \omega) + \beta \cdot Dist(v, \omega) + \gamma \cdot Vel(v, \omega)\right) \tag{9}$$

where $P(v, \omega)$ is the angular difference between the end direction of the trajectory and the current target point; $Dist(v, \omega)$ is the distance between the trajectory and the nearest obstacle; $Vel(v, \omega)$ is the current speed evaluation function; δ is the smoothing function; $\alpha, \beta, \gamma,$ are the weighting coefficients.

The traditional DWA algorithm has three main problems: Firstly, the setting of DWA weight factor is critical to completing path planning. Different DWA weight parameters can produce different planning tracks in the same environment, and identifying the optimal weight parameter setting can be challenging. Secondly, the angular velocity steering is unreasonable. Thirdly, a single DWA algorithm can easily fall into local optimal values in complex environments. To address these issues, this section proposes the following improvements: First, a fuzzy control theory-based strategy is proposed to adjust DWA weight parameters for different environments, which helps AGV navigate smoothly and safely to reach the target point. Second, introducing linear speed adjustment for angular speed makes AGV turns more reasonable. Finally, IACO and IDWA algorithms are fused to solve the problem of DWA falling into local optimal values.

3.2.1. Fuzzy control adjusts DWA parameters

In the evaluation function of traditional DWA algorithm, the three constant weight factors are generally obtained empirically and experimentally. Different combinations of these coefficients can affect DWA path planning, resulting in longer planned trajectories, unreachable targets, and collisions. Based on this analysis, this paper proposes using fuzzy control theory to adjust DWA parameters, which considers directional, safety principles and adjustment coefficients c for direction and safety. Three fuzzy controllers are designed to dynamically adjust the three weight coefficients, improving the adaptability of AGV in different environments. The fuzzy controller design is shown in Figure 4:

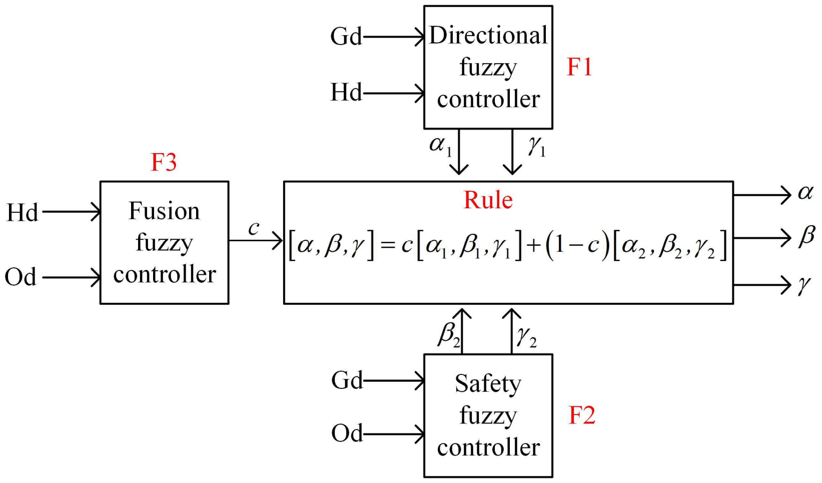


Figure 4. Block diagram of the fuzzy controller.

F1 is the directional fuzzy controller, its main influencing factors are the distance between current AGV and target point Gd , the angle between current point and target point Hd. The output includes the azimuth parameter α_1 and velocity parameter γ_1. F2 is a safety fuzzy controller, its main influencing factors are the distance between current AGV and target point Gd and the distance between current AGV and nearest obstacle Od. The output includes the obstacle parameter β_2 and speed parameter γ_2. To further increase the adaptability of DWA, a fusion fuzzy controller F3 is designed, with the main influencing factors being the distance between AGV and target azimuth Hd and the distance between current AGV and the nearest obstacle Od. The output includes the weight coefficient c, which takes into account both the direction and safety controllers. The input-output relationships of the three fuzzy controllers are shown in Figure 5.

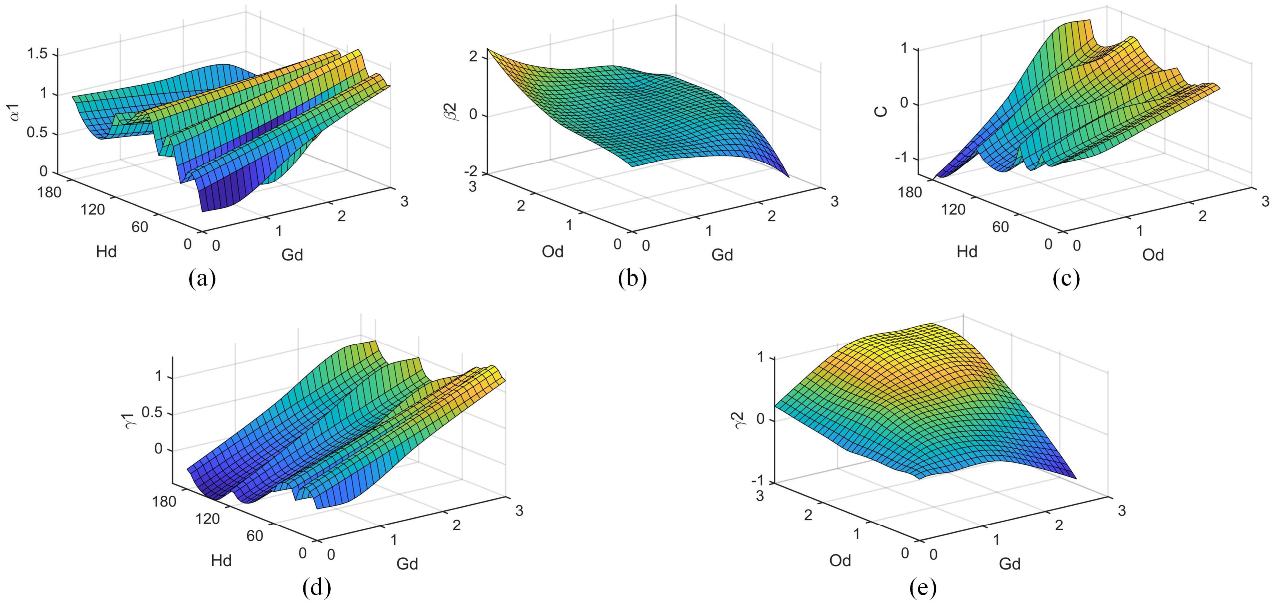


Figure 5. The input-output relations of the three fuzzy controllers.

Taking Figure 6 as an example:

Phase 1: AGV starts from the starting point. As Hd angle is small and Od distance is larger, Gd distance is also larger, causing the direction fuzzy controller to output a larger speed parameter γ_1 and an appropriate orientation angle parameter α_1. Simultaneously, safety fuzzy controller outputs a larger parameter γ_2 and a smaller obstacle parameter β_2, indicating that AGV is far from any obstacles and that direction fuzzy controller is preferred. Therefore, fusion controller outputs a larger weight coefficient c.

Phase 2: AGV runs near an obstacle. As Hd angle is large and Od distance is small, while Gd distance is still larger, direction fuzzy controller outputs an appropriate speed parameter γ_1 and a smaller orientation angle parameter α_1. Meanwhile, safety fuzzy controller outputs an appropriate parameter γ_2 and a larger obstacle parameter β_2, indicating that AGV is closer to obstacles and that safety fuzzy controller is preferred. Therefore, fusion controller should output a smaller weight coefficient c.

Phase 3: when AGV bypasses obstacles and gradually approaches target point, there are two situations. Firstly, if there is no obstacle near the target point, Hd angle is small, Od distance is larger, and the Gd distance is small. Direction fuzzy controller outputs a smaller speed parameter γ_1 and a larger orientation angle parameter α_1, while safety fuzzy controller outputs a smaller parameter γ_2 and a smaller obstacle parameter β_2. Here, AGV is closer to the target point, and direction fuzzy controller is preferred, prompting the fusion controller to output a larger weight coefficient c.Secondly, when obstacles are near the target point and the Od distance is smaller, preventing the problem of an unreachable target, direction fuzzy controller is preferred. In this case, directional fuzzy controller outputs an appropriate speed parameter γ_1 and a larger orientation angle parameter α_1, while the safety fuzzy controller outputs an appropriate speed parameter γ_2 and a suitable obstacle parameter β_2. Fusion controller outputs larger weight parameter c.

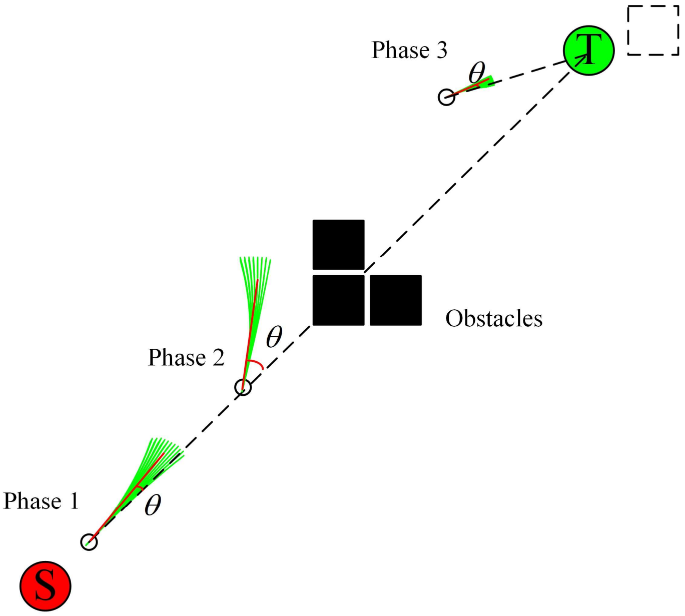


Figure 6. Schematic diagram of DWA parameter principle of fuzzy theory.

3.2.2. Correct the angular velocity

The angular velocity of traditional DWA is adjusted based on the difference between the optimal trajectory and AGV direction towards the target point. However, this fluctuation in angular velocity can lead to the AGV turning too fast, resulting in an unsatisfactory path and adversely affecting the smoothness and safety of AGV. Therefore, the angular velocity variation of AGV should be considered in environments with more obstacles. In this paper, improved angular velocity are processed by linear normalization, bringing the planned path closer to the optimal path. The improved angular velocity is shown in Eq (10):

$$\omega' = \frac{v_t}{v_{\max}} \cdot \omega_t \tag{10}$$

where v_t is the current moment linear speed, $v_{\max}$ is the maximum linear speed, and ω_t is the current angular speed.

3.3. Hybrid IACO-DWA algorithm

The hybrid algorithm is the combination of IACO and IDWA technologies. IACO algorithm can quickly obtain global path information, but the resulting path may not meet the requirements for smoothness and safety. On the other hand, IDWA algorithm has good local path planning capabilities but only uses a single final goal as a guide, making it easy to fall into local optimal.

A single path planning algorithm may not achieve path planning in complex environments, this paper combines two algorithms, using the global path obtained from IACO planning as the local target points for IDWA algorithm path planning. This ensures that IDWA algorithm for local path planning can be planned along the globally optimal path points. The proposed algorithmic framework is shown in Figure 7.

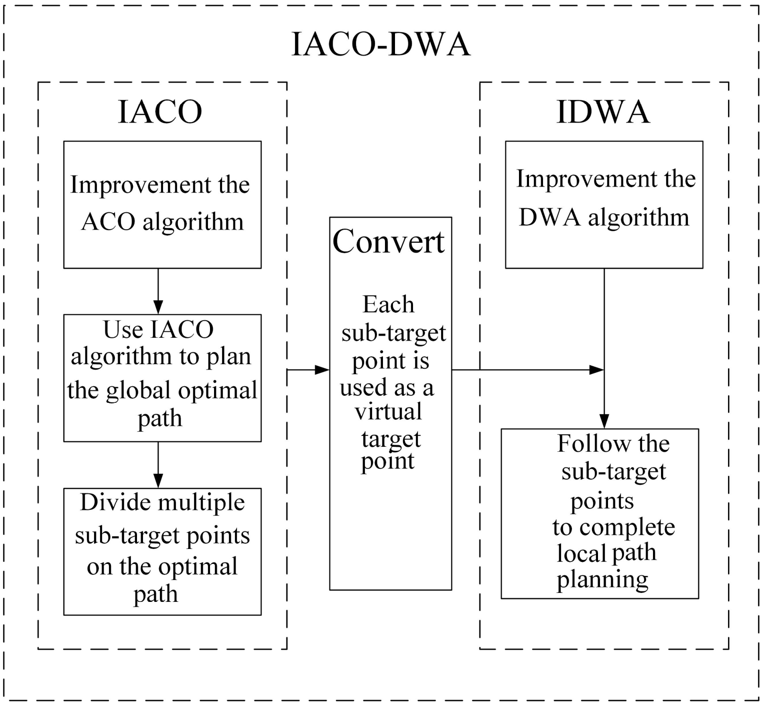

IACO-DWA
IACO
Improvement the ACO algorithm
Use IACO algorithm to plan the global optimal path
Divide multiple sub-target points on the optimal path
Convert
Each sub-target point is used as a virtual target point
IDWA
Improvement the DWA algorithm
Follow the sub-target points to complete local path planning

Figure 7. A Schematic of the path-smoothing optimization.

4. Simulation results validation

The simulation results were verified using a MATLAB R2019a simulation experiment, consisting of three parts. The first part verifies the effectiveness of IACO, while the second part verifies the effectiveness of IDWA. The last part verifies the effectiveness of IACO-DWA algorithm.

4.1. Verify the effectiveness of IACO algorithm

In the first part of the simulation experiment, Simulation 1 is adding the improved ACO strategy, IACO adds a path optimization strategy to the Simulation 1, which was compared against Simulation 1 and ACO algorithm. Figure 8 shows the comparison of optimal paths for IACO, Simulation 1, and ACO. Similarly, Figure 9 compares the optimal paths for IACO against References [23,24] (ISSA).

Figure 10 presents an iterative comparison of IACO with References [23,24] (ISSA), and ACO algorithm. The simulation data is shown in Table 1.

The simulation experiment was conducted in a grid environment of 20 × 20, with the starting and ending point coordinates set to (0.5, 19.5) and (19.5, 0.5) respectively. The number of ants used was 50, with a maximum of 100 iterations. For IACO, Simulation 1, and ACO algorithm, the parameters were set as follows: $\alpha = 1$, $\beta = 7$, $\rho = 0.3$, and $Q = 1$. The parameters for References [23,24] (ISSA) were consistent with those in the original references. Each algorithm was run 20 times, and the results were analyzed accordingly.

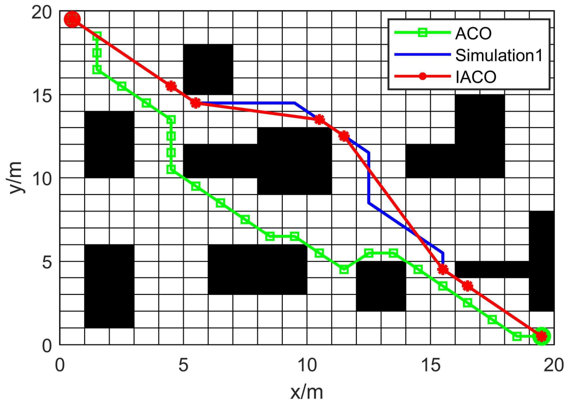


Figure 8. Optimal path comparison.

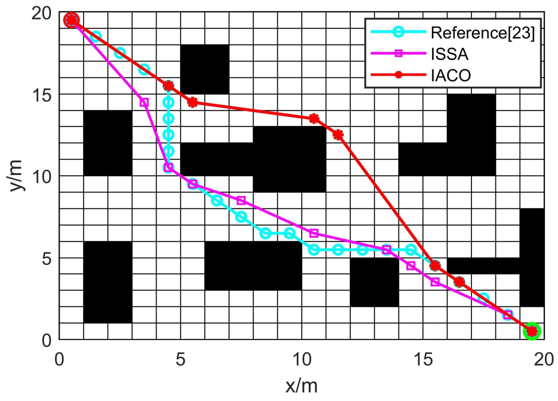


Figure 9. Optimal path comparison.

As shown in Figures 8–10, the ACO algorithm can obtain a collision-free path, but it requires improvements in terms of path quality, turn times, and iterations. Additionally, it is prone to converging to local optima. Both Reference [23] and Simulation 1 algorithms showed improvements in these aspects, with optimized path quality, turn times, and iterations. ISSA algorithm was able to obtain a shorter path and fewer turn times, but required a larger number of iterations. Finally, the IACO algorithm obtained an optimal path with optimal turn times and iteration times.

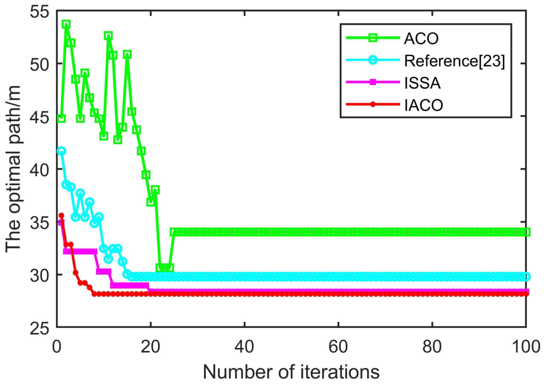


Figure 10. Algorithm iteration comparison.

Table 1. Simulation experimental data of algorithms.

Algorithm	Optimized Path length/m			Number of turns			Number of iterations		
	Best	Worst	Mean	Best	Worst	Mean	Best	Worst	Mean
ACO	30.6274	34.0416	32.0671	7	15	11.08	22	41	31
Reference [23]	29.2132	31.2132	30.1672	6	12	8.91	16	24	22
Simulation 1	29.2132	30.0416	29.6182	6	10	7.85	8	18	14
ISSA	28.3836	28.9780	28.6116	5	9	7.13	20	49	28
IACO	28.1854	28.1854	28.1854	4	4	4.00	8	17	13

According to the data presented in Table 1, IACO algorithm showed improvements in multiple aspects when compared against ACO, Reference [23], and ISSA algorithms. Specifically, the optimal path length decreased by 8.0, 3.5 and 0.7% respectively. The minimum number of turn times also decreased significantly, with reductions of 42.9, 33.3 and 20%. Additionally, the minimum iteration count decreased by 63.6, 50.0 and 60.0%. These results demonstrate the effectiveness of IACO algorithm in improving path planning in complex environments, outperforming other algorithms in terms of path quality, turn times, and iterations.

4.2. Verify the effectiveness of IDWA algorithm

To verify the effectiveness of IDWA, three different weight parameter cases were set up for traditional DWA. Two simulations were conducted for Improved DWA, one using fuzzy DWA without the addition of angular velocity improvement strategy (IDWA 1), and the other using fuzzy DWA with the added angular velocity improvement strategy (IDWA). Considering the actual radius of AGV, the AGV was considered to have reached the target point when the distance between it and the target was less than 0.5 m. The weight parameters for DWA are shown in Table 2, while IDWA 1 and IDWA parameters were set as shown in Figure 5. The AGV kinetic parameters were set as shown in Table 3. In order to consider both security and path length, simulation data was analyzed based on these indicators, as shown in Table 4. Figure 11 shows the comparison of paths generated by the three different weight parameters and IDWA algorithm. Figure 12 shows the weight parameter change curves for IDWA path planning, while Figure 13 shows a comparison of linear velocity and angular velocity between IDWA and DWA. These results were used to verify the effectiveness of IDWA algorithm.

Table 2. Weight parameter values of DWA.

Case	α	β	γ
Case1	0.2	0.1	0.1
Case2	0.1	0.3	0.3
Case3	0.1	0.1	0.1

Table 3. Kinetic parameter values of AGV.

Parameter	Value
start	(0,0)
target	(8,8)
v_{max}	0.5 m/s
v_a	0.2 m/s²
ω_{max}	20°/s
ω_a	50°/s²
Linear Velocity Resolution	0.01 m/s
Angular velocity resolution	1°/s
Time resolution	0.1 s
Forecast period	3 s

Table 4. Comparison of simulation data of DWA and IDWA.

Simulation	path	safty
Case1	–	No
Case2	–	Yes
Case3	11.7440	No
IDWA 1	11.6330	Yes
IDWA	11.6320	Yes

As shown in Figure 11 and Table 4, the path planned by Case1 resulted in a collision and did not meet the safety indicators. While the paths generated by Case2 and Case3 met safety requirements, they were longer than desired, failing to meet the path shortening indicators. However, after IDWA added the angular velocity improvement strategy, the resulting path was shorter than IDWA1, demonstrating the effectiveness of this strategy. Moreover, the path planned using IDWA was better, thus validating the effectiveness of IDWA algorithm for path planning in complex environments.

Figure 12 shows that during the early stages of AGV operation, where there are fewer obstacles present, larger values of α and γ can be used in IDWA, allowing AGV to maintain higher speeds.

As the AGV approaches an obstacle, parameter β is increased. Parameters α, β, and γ are adjusted accordingly based on changes in the environment to ensure that AGV can safely reach its target point. This highlights the adaptability of IDWA algorithm to different environments, allowing it to effectively plan paths and adjust its weight parameters based on real-time conditions. As shown in Figure 13(a), the line velocity of IDWA can reach the set line speed value with a large linear speed, while DWA is unable to maintain high line velocity due to fixed weight parameters and changes in AGV environment. Similarly, Figure 13(b) shows that the angular velocity variation interval of IDWA is small, and the number of fluctuations is also small, indicating that the turning amplitude of AGV is small when running, further demonstrating the effectiveness and efficiency of IDWA algorithm in dealing with complex environments.

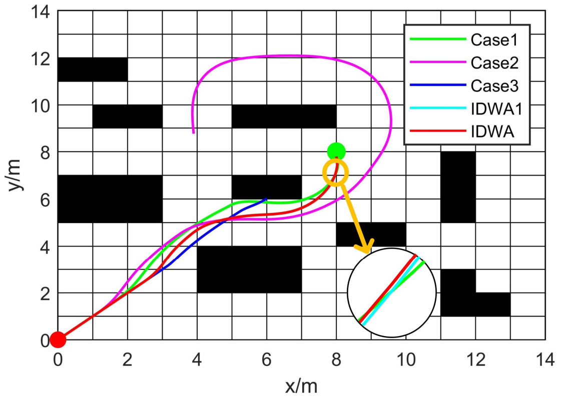


Figure 11. Path comparison of IDWA and DWA.

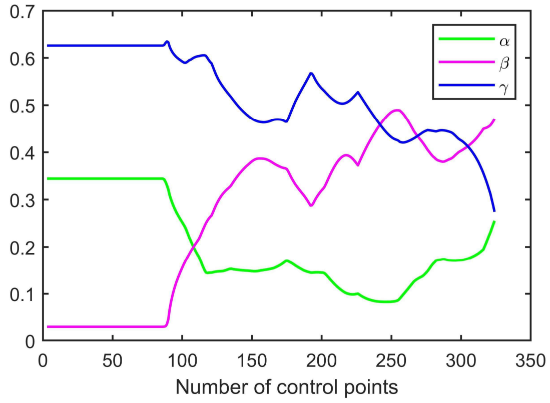


Figure 12. The weight parameter of IDWA.

The simulation results demonstrate that the weight parameter settings for DWA can significantly affect path planning efficiency, with fixed weight parameters poorly adapting to complex environments. In complex obstacle environments or when obstacles are present at target points, inappropriate weight parameter settings may result in collisions and long, non-smooth paths. By adding the angular velocity improvement strategy, IDWA was able to further optimize its path planning. Moreover, IDWA is able to dynamically adjust algorithm weight parameters, exhibiting strong adaptability to AGV environments and generating paths that are both safer and more efficient.

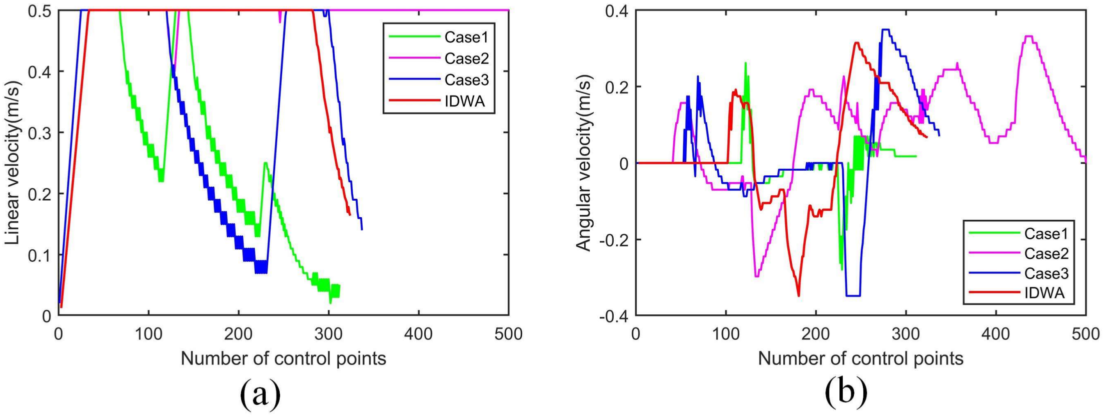


Figure 13. Comparison of linear velocity and angular velocity between IDWA and DWA.

4.3. Verify the effectiveness of IACO-DWA algorithm

This section presents a simulation of ISSA, Reference [25,26] and IACO-DWA algorithms. The IACO-DWA algorithm parameters are set as the same as in the previous section, and the parameters for ISSA, Reference [25,26] are consistent with those specified in the original reference. In consideration of AGV working characteristics in real environments, four evaluation indicators were used in this simulation experiment: optimized path length, number of turns, safety, and smoothness. Table 5 shows the simulation data for the four algorithms, with start coordinates set to (0.5, 0.5) and end coordinates set to (19.5, 16.5). Figure 14 presents a comparison of optimal paths generated by the four algorithms. Figure 15 shows the weight parameter changes for IACO-DWA algorithm, while Figure 16 compares the linear and angular velocity between IACO-DWA and Reference [25].

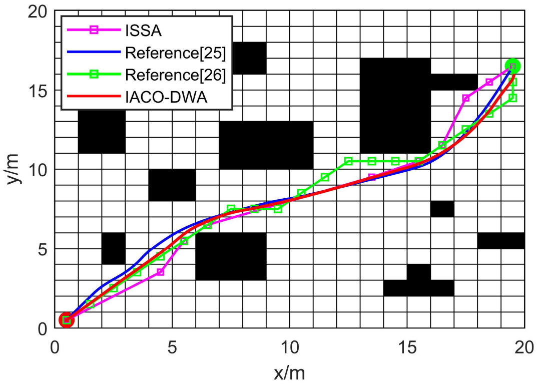


Figure 14. Path comparison of the optimal path.

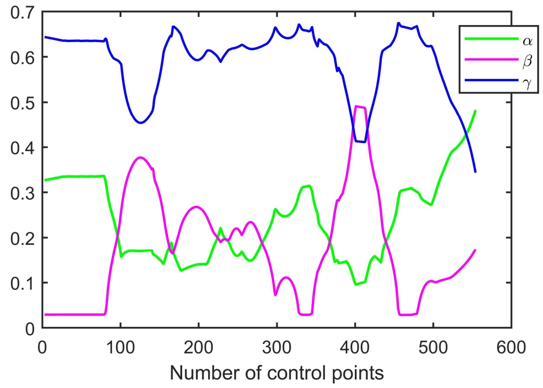
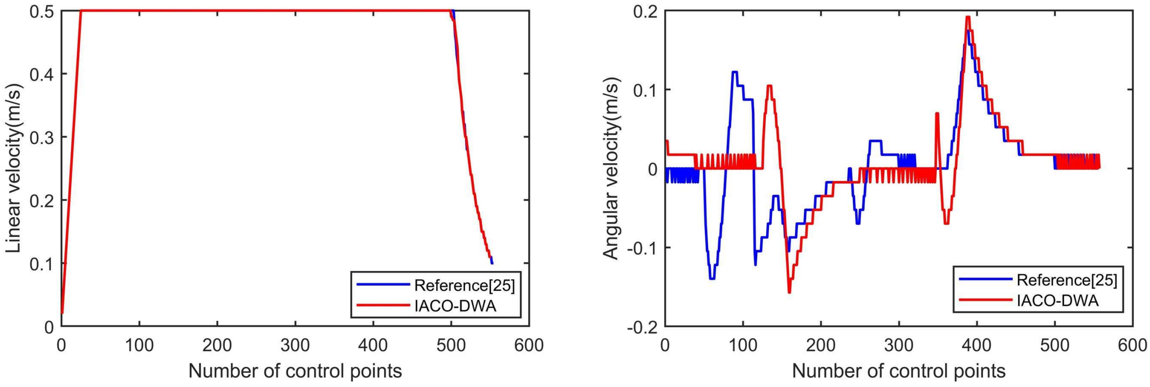


Figure 15. The weight parameter of IACO-DWA.



(a) (b)

Figure 16. Comparison of linear velocity and angular velocity between IACO-DWA and Reference [25].

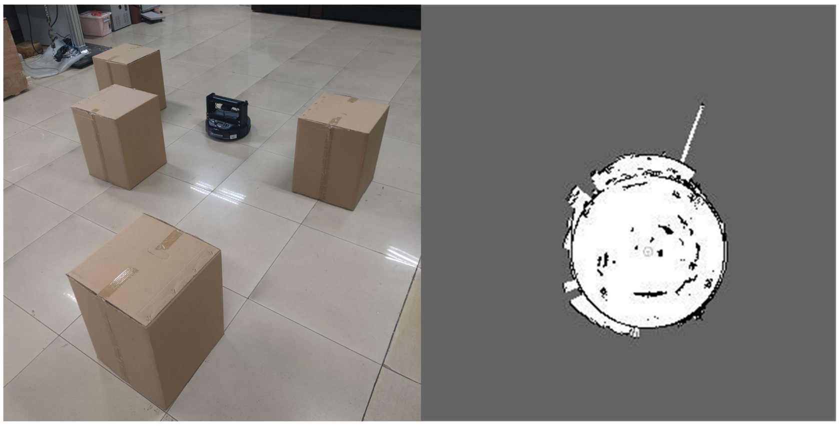
Table 5. Simulation experimental data of algorithms.

algorithm	Optimized Path length/m	Number of turns	Smoothness	Safty
ISSA	25.9070	6	No	No
Reference [25]	26.0660	2	Yes	Yes
Reference [26]	26.7990	5	No	No
IACO-DWA	25.5070	2	Yes	Yes

As shown in Figure 14 and Table 5, ISSA algorithm and Reference [26] is capable of generating a collision-free path, but the path length is longer, the number of turns is too much, and the path is not smooth or safe enough. However, both Reference [25] and IACO-DWA algorithms are able to generate better paths that meet safety and smoothness indicators while maintaining a safe distance from obstacles to prevent collisions. Compared with the above three algorithms, IACO-DWA algorithm was able to reduce optimal path length by 1.6, 2.2 and 5.1%. Weight parameter curves for IACO-DWA were adjusted based on changes in the environment, as demonstrated in Figure 15. Furthermore, Figure 16(a) shows that both Reference [25] and IACO-DWA algorithms can maintain high linear speeds, while Figure 16(b) demonstrates that angular velocity fluctuations in IACO-DWA are better than Reference [25]. These results demonstrate the effectiveness of IACO-DWA algorithm in generating optimal paths that allow AGV to safely and smoothly reach their target points while also adapting to changing environments.

5. QBot2e experimental validation

This section presents a verification of the effectiveness and feasibility of IACO-DWA algorithm in practical indoor AGV testing. The experimental setup used a two-wheel-drive differential mobile robot called QBot2e, which is an ground mobile robot produced and developed by Quanser. QUARC is a software model developed by Quanser company to control the QBot2e robot in real time. This software model is seamlessly integrated with Matlab/Simulink, allowing users to write the corresponding program directly using Matlab/Simulink, which can control the movement of the QBot2e robot. QBot2e is equipped with Raspberry Pi and a 640 × 480 Microsoft vision Kinect sensor to collect information about the external environment in a range of 0.5–6 meters. To demonstrate the effectiveness of IACO-DWA algorithm, this section is based on Matlab / Simulink software and Quanser control model, ISSA, Reference [25,26], and IACO-DWA algorithms are verify the effectiveness of the IACO-DWA algorithm by the path planned by QBot2e. Table 6 is the data of the four algorithms in the real environment. The algorithm parameters agree with the simulation experiments and the experimental environment consists of four cartons of 0.5 × 0.5 × 1.0 m^3, Figure 17(a) is the placement diagram of indoor obstacles, Figure 17(b) is the infographic diagram scanned by QBot2e, Figure 18 is a comparison diagram of the optimal paths of the four algorithms Figure 19 is an effect diagram of IACO-DWA algorithm running, Figure 19(a) is the starting position of QBot2e, (b) is QBot2e at the first turn, (c),(d) is the QBot2e through the obstacle scene, (e) is the QBot2e reaching the second turn, (f) is the QBot2e reaches the ending point. Table 6 is the experimental data of the four algorithms in a real environment. The experimental data parameters were set as in Section 4.3. The experiment was conducted through the following three steps:



(a) (b)

Figure 17. Indoor obstacle environment map**.**

Step 1: Control the QBot2e robot in Simulink to scan the obstacle environment, sense the obstacle information through the on-board Kinect sensor, and generate the obstacle environment map, as shown in Figure 17(b).

Step 2: Convert the obstacle map information of environment into a grid map through image processing technology in Matlab, set the starting point and ending point on the grid map, starting point and ending point are indicated by red and green dots, respectively. Running the four algorithms via Matlab generated a path comparison map of the four algorithms as shown in Figure 18.

Step 3: Sent the path points generated by the IACO-DWA algorithm to QBot2e. The path point was compared to the current location of the QBot2e. Use a PID controller in Simulink until the QBot2e reached the ending point. The QBot2e travels according to the planned path, as shown in Figure 19.

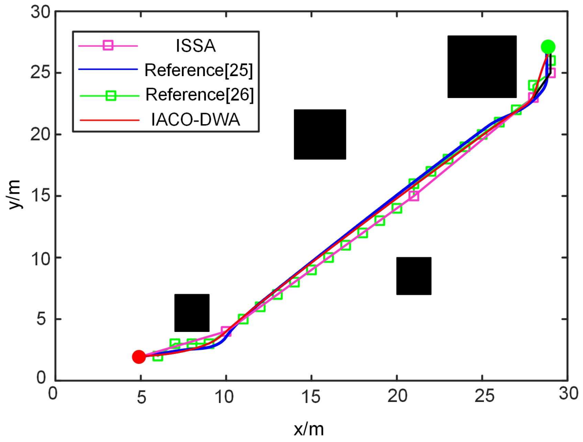
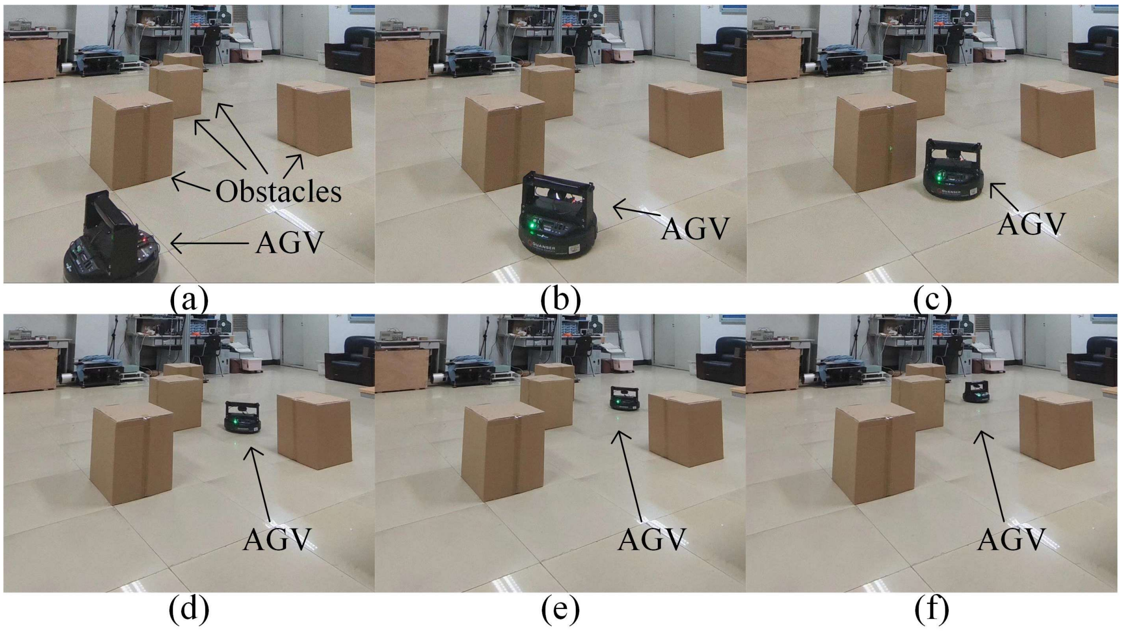


Figure 18. Comparison of optimal paths in indoor environment.



(a) (b) (c)

(d) (e) (f)

Figure 19. Indoor experiment display.

Table 6. Experimental data of algorithms.

algorithm	Optimized Path length/m	Number of turns	Smoothness	Safty
ISSA	3.508	4	No	Yes
Reference [25]	3.539	2	Yes	Yes
Reference [26]	3.570	7	No	Yes
IACO-DWA	3.482	2	Yes	Yes

Figure 18 and Table 6 can be seen that ISSA and Reference [26] has more turns and longer paths, and their path is not smooth enough. Compared with the existing fusion algorithm Reference [25], IACO-DWA algorithm has a better path, the path length was reduced by 1.6%. In addition, the path planned by the IACO-DWA algorithm has a certain safe distance between the obstacle on the premise of satisfying the smoothness, which reduces the risk of collision between QBot2e and the obstacle in the process of movement. The above experimental results show that the practicability and effectiveness of IACO-DWA algorithm are validated through the QBot2e platform.

6. Conclusions and future works

In this paper, a new IACO-DWA algorithm based on improved ACO and DWA algorithms was proposed for path planning of AGV. The effectiveness of IACO-DWA was verified through simulations and testing on the QBot2e platform. The improved ACO algorithm was used to generate the optimal path, while the improved DWA algorithm was used for local path planning, enabling AGV safely and smoothly to reach their target points. Simulation experiments were conducted to verify the validity of IACO-DWA algorithm, and the utility of the algorithm was further demonstrated through testing on the QBot2e platform. Overall, IACO-DWA algorithm proved effective in generating optimal paths for AGV in complex indoor environments, highlighting its potential for practical application.

The limitation of the algorithm in this paper are currently only applicable to the static obstacle environment. In future work, further research is needed to investigate dynamic environments, including path planning when facing dynamic obstacles. In such dynamic environments, the path planning and collision avoidance strategies of AGV will be a major focus of future research. Moreover, it is necessary to verify the practical application of the IACO-DWA algorithm in a dynamic obstacle environment.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant (62103127，62103126), Natural Science Foundation of Hebei Province under Grant F2020201048, Funded by Science and Technology Project of Hebei Education Department under Grant BJK2023057, Advanced

Talents Incubation Program of the Hebei University under Grant 521000981366 and Foundation of President of Hebei University under Grant XZJJ201906.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. L. Zhang, W. Hu, B. Kang, J. Wang, Y. Lu, Automatic assessment of depression and anxiety through encoding pupil-wave from HCI in VR scenes, *ACM Trans. Multimedia Comput. Commun. Appl.*, **2022** (2022). https://doi.org/10.1145/3513263
2. L. Liu, X. Wang, X. Yang, H. Liu, J. Li, P Wang, Path planning techniques for mobile robots: Review and prospect, *Expert Syst. Appl.*, **227** (2023), 120254. https://doi.org/10.1016/j.eswa.2023.120254
3. D. Bechtsis, N. Tsolakis, D. Vlachos, E. Iakovou, Sustainable supply chain management in the digitalisation era: The impact of Automated Guided Vehicles, *J. Clean. Prod.*, **142** (2017), 3970–3984. https://doi.org/10.1016/j.jclepro.2016.10.057
4. S. Wu, Y. Du, Y. Zhang, Mobile robot path planning based on a generalized wavefront algorithm, *Math. Probl. Eng.*, **2020** (2020), 1–12. https://doi.org/10.1155/2020/6798798
5. S. Anthony, The Focussed D* Algorithm for Real-Time Replanning, *Proc. Int. Joint Conf. Artif. Intell.*, **1** (2002), 968–975. https://doi.org/10.1109/ROBOT.2002.1013481
6. J. J. Kuffner, S. M. LaValle, RRT-connect: An efficient approach to single-query path planning, *IEEE Int. Conf. Robot.*, **2** (2000), 995–1001. https://doi.org/10.1109/ROBOT.2000.844730
7. H. Chen, T. Wang, T. Chen, W. Deng, Hyperspectral image classification based on fusing S3-PCA, 2D-SSA and random patch network, *Remote. Sens.*, **15** (2023), 3402. https://doi.org/10.3390/rs15133402
8. G. Sayed, M. Soliman, A. Hassanien, A novel melanoma prediction model for imbalanced data using optimized SqueezeNet by bald eagle search optimization, *Comput. Biol. Med.*, **136** (2021), 104712. https://doi.org/10.1016/j.compbiomed.2021.104712
9. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future. Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
10. G. G. Wang, G. S. Hao, S. Cheng, Q. Qin, A discrete monarch butterfly optimization for Chinese TSP problem, *Lect. Notes Comput. Sci.*, **9712** (2016). https://doi.org/10.1007/978-3-319-41000-5_16
11. K. Ong, C. Sia, A carnivorous plant algorithm for solving global optimization problems, *Appl. Soft. Comput.*, **98** (2021), 106833, https://doi.org/10.1016/j.asoc.2020.106833
12. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst.*, **26**, (1996), 29–41. https://doi.org/10.1109/3477.484436
13. Y. Zhou, N. Huang, Airport AGV path optimization model based on ant colony algorithm to optimize Dijkstra algorithm in urban systems, *Sustain. Comput. Inf.*, **35** (2022), 100716. https://doi.org/10.1016/j.suscom.2022.100716

14. Q. Luo, H. Wang, Z. Yan, J. He, Research on path planning of mobile robot based on improved ant colony algorithm. *IEEE. Trans. Neural. Network Learn. Syst.*, **32** (2020), 1555–1566. https://doi.org/10.1007/s00521-019-04172-2
15. L. Yang, L. Fu, P. Li, J. Miao, N. Guo, LF-ACO: an effective formation path planning for multi-mobile robot. *Math. Biosci. Eng.*, **19** (2022), 225–252. https://doi.org/10.3934/mbe.2022012
16. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *IEEE Int. Conf. Robot.*, **1985** (1985), 500–505. https://doi.org/10.1109/ROBOT.1985.1087247
17. D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, *IEEE Robot. Autom. Mag.*, **4** (1997), 23–33. https://doi.org/10.1109/100.580977
18. L. Chang, L. Shan, C. Jiang, Y. Dai, Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment, *Auton. Robot.*, **45** (2021), 51–76. https://doi.org/10.1007/s10514-020-09947-4
19. S. Han, L. Wang, Y. Wang, H. He, A dynamically hybrid path planning for unmanned surface vehicles based on non-uniform Theta* and improved dynamic windows approach, *Ocean. Eng.*, **257** (2022), 111655. https://doi.org/10.1016/j.oceaneng.2022.111655
20. S. Wang, Y. Hu, Z. Liu, L. Ma, Research on adaptive obstacle avoidance algorithm of robot based on DDPG-DWA, *Comput. Electron. Eng.*, **109** (2023), 108753. https://doi.org/10.1016/j.compeleceng.2023.108753
21. X. Bai, B. Li, X. Xu, Y. Xiao, USV path planning algorithm based on plant growth. *Ocean. Eng.*, **273** (2023), 113965. https://doi.org/10.1016/j.oceaneng.2023.113965
22. X. Tian, L. Liu, S. Liu, Z. Du, M. Pang, Path planning of mobile robot based on improved ant colony algorithm for logistics, *Math. Biosci. Eng.*, **18** (2021), 3034–3045. https://doi.org/10.3934/mbe.2021152
23. C. Miao, G. Chen, C. Yan, Y. Wu, Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, *Comput. Ind. Eng.*, **156** (2021), 107230. https://doi.org/10.1016/j.cie.2021.107230
24. Z, Zhang, R. He, K. Yang, A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. *Adv. Manuf.*, **10** (2022), 113–130. https://doi.org/10.1007/s40436-021-00366-x
25. X. Ji, S. Feng, Q. Han, H. Yin, S. Yu, Improvement and fusion of A* algorithm and dynamic window approach considering complex environmental information, *Arab. J. Sci. Eng.*, **46** (2021), 7445–7459. https://doi.org/10.1007/s13369-021-05445-6
26. L. Wu, X. Huang, J. Cui, C. Liu, W. Xiao, Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot, *Expert Syst. Appl.*, **215** (2023), 119410. https://doi.org/10.1016/j.eswa.2022.119410